

Open Source AI

Projects, Insights, and Trends

By Ibrahim Haddad, Ph.D.

Table of Contents

Introduction

Surveyed Projects

12

Projects

TensorFlow

18

Language Breakdown

20

GitHub Project Insights

21

Observations

22

Microsoft Cognitive Toolkit (CNTK)

28

Language Breakdown

30

GitHub Project Insights

31

Observations

32

Convolutional Architecture for Fast Feature Embedding (Caffe)

36

Language Breakdown

38

GitHub Project Insights

39

Observations

40

Caffe2

44

Language Breakdown

46

GitHub Project Insights

47

Observations

48

Theano	50
Contribution History	51
Language Breakdown	53
GitHub Project Insights	54
Observations	55
Torch	56
Language Breakdown	57
GitHub Project Insights	57
Contribution History	58
Accord.NET	60
Language Breakdown	61
GitHub Project Insights	62
Contribution History	63
Apache SINGA	66
Language Breakdown	68
GitHub Project Insights	68
Observations	69
Apache Mahout	70
Languages Breakdown	72
Contribution History	72
Apache Spark	76
Languages Breakdown	78
Observations	79
Eclipse DeepLearning4J (DL4J)	80
Language Breakdown	82

GitHub Project Insights	82
Observations	83
Keras	86
Language Breakdown	88
GitHub Project Pulse	88
Observations	89
Apache MXNet	92
Language Breakdown	94
GitHub Project Insights	95
Observations	96
Apache PredictionIO	98
Language Breakdown	99
Observations	99
Apache SystemML	102
Language Breakdown	104
Observations	105
The Linux Foundation AI Efforts	108
Observations	111
Common Characteristics	112
Development and governance is dominated by a single large entity	112
Developed internally to address specific product requirements	112
Highly specialized to perform certain types of tasks	113
Little contributor cross-pollination	113
Most projects are tightly coupled with their original authors	113
Fast Release Cycles Dominate Development	114

Major Improvements in Documentation	114
Academia and AI	115
Open Source and AI	122
Exodus from Academia	122
Incubators Work	122
GitHub is Dominant	123
Open Source Licenses and Open Training Data	123
Consolidation, Winners and Winners	124
Governance Matters	124
Open Source Development and Collaboration Model	124
Access to a Larger User and Developer Ecosystem	125
Improved Code Quality and Stability	125
Create Technical and Political Leadership	125
Reduces Licensing Costs	125
Collaboration	126
Faster Innovation	126
Faster Time to Market	126
Appendix A – Methodology of Contribution Analysis	127
Tool	127
Characterization of the Results	127
Development Statistics	128

Acknowledgments	129
Feedback	129
About the Author	130

SECTION I

Introduction

Surveyed Projects

An ever-growing number of open source projects in the AI domain are aiming to solve various types of problems and offer solutions at the platform level (cluster computing frameworks), libraries and frameworks for machine and deep learning, libraries to evaluate mathematical expressions, neural network libraries, platforms for training models over large datasets, and more.

In this guide, we explore 16 such open source projects. We provide summarized information about the projects, and explore their development statistics to gain valuable insights. Writing such a guide is a great learning experience that allows one to observe trends and identify insights about the projects and the domain in general. In the concluding section, we present some observations that touch on the role of the open source development, common characteristics among the surveyed projects, what is happening in academia, and why everybody is a winner!

The initial challenge for writing this ebook was identifying where to start and what projects to feature. We started by looking at popular, well-known projects and decided to capture and document projects as we discover them within that ecosystem of open source AI. The result is an initial batch of 16 projects listed below. It is hard to profile all open source AI projects in one publication. We plan to introduce other projects over

multiple installments. If you would like to see other projects featured, please contact the author via <http://www.ibrahimatlinux.com/contact.html>.

For each of these initial projects, we provide a section that presents the project in a summarized format that offers basic, must-know information and pointers to the project's web and code resources. We also provide information on their GitHub presence and statistics about their development efforts using **Facade**. The development statistics include the number of commits, number of unique contributors, number of added LoC, number of removed LoC, etc. At the end of each project's section, we offer a set of observations based on the data we collected and analyzed.

Project	Short Description	License
TensorFlow	Software library for numerical computation using data flow graphs.	Apache 2.0
Microsoft Cognitive Toolkit (CNTK)	Deep learning framework that describes neural networks as a series of computational steps via a directed graph.	MIT
CAFFE	Deep learning framework, originally developed at UC Berkeley, supports many different types of deep learning architectures geared towards image classification and image segmentation.	BSD 2-Clause
CAFFE2	Deep learning framework made for expression, speed, and modularity. It has been adapted to many machine learning functions.	Apache 2.0
Theano	Python library that allows its users to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.	BSD 3-Clause
Torch	Simple framework for building complex scientific algorithms to handle computer vision, signal processing, parallel processing, image, video, and networking.	BSD 3-Clause

Project	Short Description	License
Accord.NET	Machine learning framework combined with audio and image processing libraries completely written in C#.	LGPL 2.1
Apache Mahout	Implementations of distributed, scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification.	Apache 2.0
Apache Spark	A cluster-computing framework for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing.	Apache 2.0
DeepLearning4J	Deep learning programming library (written for Java and the Java virtual machine) and a computing framework with wide support for deep learning algorithms.	Apache 2.0
Keras	Neural network library written in Python. It is capable of running on top of TensorFlow, CNTK, Theano, or Apache MXNet. Keras is designed to enable fast experimentation with deep neural networks.	MIT

Project	Short Description	License
Apache MXNet	Deep learning framework used to train, and deploy deep neural networks. It is scalable, allowing for fast model training, and supports a flexible programming model and multiple languages.	Apache 2.0
Apache SINGA	General distributed, deep learning platform for training deep learning models over large datasets.	Apache 2.0
Apache PredictionIO	Machine learning server built on an open source stack that enables developers to manage and deploy production-ready predictive services for various kinds of machine learning tasks.	Apache 2.0
Apache SystemML	A flexible machine learning system that automatically scales to Spark and Hadoop clusters.	Apache 2.0
Acumos	Acumos AI is a platform and open source framework that makes it easy to build, share, and deploy AI apps.	Apache 2.0



SECTION II

Projects

TensorFlow

Project Creator	Google
Description	TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.
Project History	Researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization originally developed TensorFlow™ for the purposes of conducting machine learning and deep neural networks research. Google released TensorFlow under the Apache 2.0 open source license on November 9, 2015.
TensorFlow Lite	TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices. It enables on-device machine learning inference with low latency and a small binary size. TensorFlow Lite also supports hardware acceleration with the Android Neural Networks API.
Current Status	Active development. Last release was version 1.6.0 on February 28, 2018. At the time of writing, the project is on release 1.7.0-rc0.
Releases	The project has gone through 52 releases since it landed on GitHub. There is very good release notes explaining major features, improvements, bug fixes and other changes. Release note also have attributions to all contributors.

Roadmap	The project maintains a roadmap that highlights the list of priorities and focus areas of the core set of TensorFlow developers and lists the functionality to be expected in the upcoming releases. https://www.tensorflow.org/community/roadmap
License	Apache 2.0
Web Site	https://www.tensorflow.org/
Blog	https://research.googleblog.com/search/label/TensorFlow
Code Repository	https://GitHub.com/tensorflow/tensorflow
Twitter	@TensorFlow
Platforms	Linux, macOS, Windows
APIs	<p>TensorFlow has APIs available in several languages for both constructing and executing a TensorFlow graph. The Python API is the most complete and the easiest to use, but other language APIs may be easier to integrate into projects and may offer some performance advantages in graph execution. APIs in languages other than Python are not yet covered by the API stability promises (those include APIs in C++, Java, and Go).</p> <p>The project encourages its community to develop and maintain support for other languages and offers a documented approach on how to do so as recommended by the TensorFlow maintainers.</p>

Language Breakdown































Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage	
 C++	732.638	153.690	17.3%	136.238	1,022.566		46.5%
 Python	509.264	198.937	28.1%	125.236	833.437		37.9%
 HTML	252.319	46	0.0%	5.853	258.218		11.8%
 Go	16.708	14.572	46.6%	1.983	33.263		1.5%
 Java	10.861	4.187	27.8%	2.355	17.403		0.8%
 shell script	7.199	3.747	34.2%	1.730	12.676		0.6%
 CMake	4.056	1.026	20.2%	544	5.626		0.3%
 C	3.276	2.810	46.2%	1.248	7.334		0.3%
 Objective-C	2.050	336	14.1%	405	2.791		0.1%
 XML	1.314	482	26.8%	278	2.074		0.1%
 Make	1.006	265	20.8%	192	1.463		0.1%
 DOS batch script	176	112	38.9%	83	371		0.0%
 Perl	150	41	21.5%	36	227		0.0%
 JavaScript	15	11	42.3%	5	31		0.0%
 Autoconf	10	0	0.0%	5	15		0.0%
Totals	1,541,042	380,262		276,191	2,197,495		

Figure 1: TensorFlow is written primarily in C++ (46.5%) and Python (37.9%)

February 20, 2018 - March 20, 2018

Period: 1 month

Overview

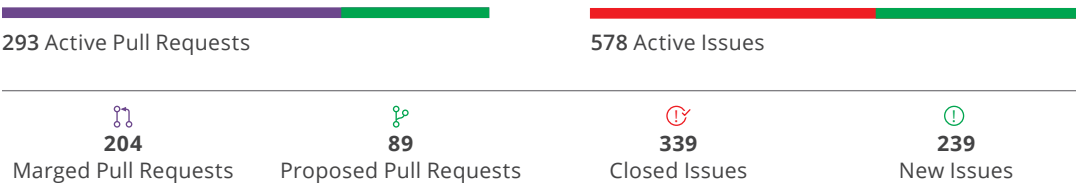


Figure 2: TensorFlow - GitHub project pulse for the period from February 20 to March 20, 2018

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

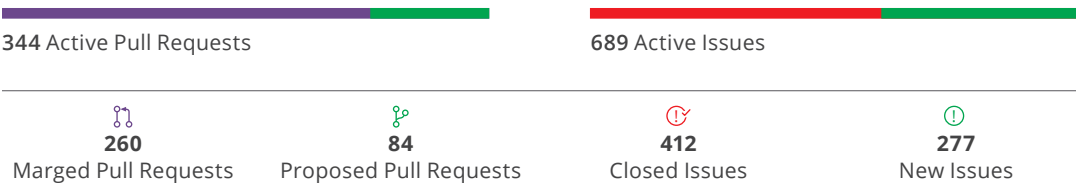


Figure 3: TensorFlow - GitHub project pulse for the period from March 15 to April 15, 2018

Figure 2 displays the project’s GitHub pulse for 1 month, February 20 to March 20, 2018.

Excluding merges, 175 authors have pushed 1,229 commits to master and 1,260 commits to all branches. On the master branch, 2,357 files have changed and there have been 114,595 additions and 54,355 deletions. Finally, 339 issues were closed and 239 new issues emerged.

We came back to check on the project three weeks later and the stats are still very impressive as illustrated in Figure 3. Excluding merges, 188 authors have pushed 1,348 commits to master and 1,393 commits to all branches. On master, 2,730 files have changed and there have been 184,388 additions and 64,422 deletions. Finally, 412 issues were closed and 277 new issues emerged.

Observations

Large and active development team

Since it became an open source project, TensorFlow has attracted contributions from over 1600 unique contributors with a very clear upward trend. This is a very large open source team. Figure 4 lists all the contributors to the project sorted by affiliation.

	2013	2014	2015	2016	2017	2018	Total
Google	0	0	15	131	185	155	254
(Unknown)	0	0	37	436	637	242	1.210
Intel	0	0	0	3	16	13	22
Infobox	0	0	0	0	1	1	1
(Hobbyist)	0	0	0	10	9	3	15
NVidia	0	0	0	0	3	5	6
Codeplay	0	0	0	1	3	1	3
(Academic)	0	0	0	22	35	8	61
Uptake	0	0	1	1	0	0	1
Microsoft	0	0	0	3	7	0	7
Graphcore	0	0	0	0	1	1	1
IMB	0	0	0	1	10	7	14
minds.ai	0	0	0	0	1	0	1
Mellanox	0	0	0	0	3	0	3
Apache Software Foundation	0	0	1	2	0	0	2
Huawei	0	0	0	2	0	0	2
Winton (Global Investment Management)	0	0	0	0	0	1	1
Yahoo!	0	0	0	4	8	2	12

Figure 4: TensorFlow - unique contributors, by affiliation (2018 covers January to March 26)

Number 42	0	0	0	0	1	0	1
Spreadtrum Communication	0	0	0	0	1	1	1
NAVER	0	0	0	2	2	0	4
H2O.ai	0	0	0	0	0	1	1
RStudio	0	0	0	0	1	0	1
Yandex	0	0	0	1	0	1	2
Clarifai	0	0	0	1	0	0	1
Amazon	0	0	0	0	0	1	1
Alibaba Cloud	0	0	0	0	1	0	1
Baidu	0	0	0	1	1	0	2
Caicloud	0	0	0	0	1	1	1
CERN	0	0	1	0	0	0	1
Institute Eldorado	0	0	0	0	1	0	1
Samsung	0	0	0	0	1	0	1
Mozilla	0	0	0	0	0	1	1
UBER	0	0	0	0	1	1	1
China Mobile	0	0	0	0	1	0	1
ZTE	0	0	0	0	1	0	1
AMD	0	0	0	0	0	1	1
Senseta	0	0	0	0	0	1	1
Shopify	0	0	0	0	0	1	1
SINA	0	0	0	0	1	0	1
Numenta	0	0	0	1	0	0	1

Figure 4: TensorFlow - unique contributors, by affiliation (2018 covers January to March 26)

Increasing Y-O-Y development activity

Project Creator	2015	2016	2017	2018
Added LoC	412,534	1,338,629	2,169,572	889,577
Removed LoC	33,750	840,626	1,220,589	387,239
Patches	528	12,033	14,007	5,046
Unique Contributors	55	622	933	449

Figure 5: Project metrics by year from January 1, 2015 to April 15, 2018

Figure 5 illustrates the year-over-year activities in four different metrics, and we can see a substantial increase in activity by all metrics. These indicate that interest in this project is rising, and that the open source and enterprise community at large has embraced this project by deploying and customizing it for their own needs and requirements.

Established codebase

Despite the relative young age of the project, the current code base is well established and used by over 90 companies including NVidia, UBER, Dropbox, eBay, Google, Snapchat, Intel, Qualcomm, Twitter, ARM, Lenovo and many more. We also see these same companies contributing to the source code base.

Figure 6 displays the history of commits to the project by authors' affiliations. The heavy contributions and technical leadership of Google has not been a deterrent for other enterprise players to get involved, contribute to, and adopt TensorFlow. However, it is very apparent that the majority of the enterprise development comes from Google even though we see contributions from dozens of companies in smaller chunks. Most likely, these contributions are minor improvements and adding functionalities to meet the requirement of specific usage model.

	2013	2014	2015	2016	2017	2018	Total
Google	0	0	413	10.306	11.677	4.002	26.398
(Unknown)	0	0	63	1.323	1.569	674	3.629
Intel	0	0	0	4	132	31	167
Infobox	0	0	0	0	196	140	336
(Hobbyist)	0	0	0	34	39	11	84
NVIDIA	0	0	0	0	13	90	103
Codeplay	0	0	0	51	57	1	109
(Academic)	0	0	0	58	125	42	225
Uptake	0	0	49	136	0	0	185
Microsoft	0	0	0	40	59	0	99
Graphcore	0	0	0	0	22	5	27
IMB	0	0	0	2	47	14	63
minds.ai	0	0	0	0	7	0	7
Mellanox	0	0	0	0	12	0	12
Apache Software Foundation	0	0	1	33	0	0	34
Huawei	0	0	0	15	0	0	15
Winton (Global Investment Management)	0	0	0	0	0	1	1
Yahoo!	0	0	0	11	22	18	51
Number 42	0	0	0	0	1	0	1
Spreadtrum Communication	0	0	0	0	3	1	4
NAVER	0	0	0	16	5	0	21
H2O.ai	0	0	0	0	0	7	7
RStudio	0	0	0	0	7	0	7
Yandex	0	0	0	1	0	1	2
Clarifai	0	0	0	1	0	0	1
Amazon	0	0	0	0	0	1	1

Figure 6: TensorFlow - patches by authors affiliation (2018 covers January to March 26)

Alibaba Cloud	0	0	0	0	4	0	4
Baidu	0	0	0	1	1	0	2
Caicloud	0	0	0	0	1	1	2
CERN	0	0	1	0	0	0	2
Institute Eldorado	0	0	0	0	1	0	1
Samsung	0	0	0	0	1	0	1
Mozilla	0	0	0	0	0	1	1
UBER	0	0	0	0	1	1	2
China Mobile	0	0	0	0	1	0	1
ZTE	0	0	0	0	3	0	3
AMD	0	0	0	0	0	1	1
Senseta	0	0	0	0	0	1	1
Shopify	0	0	0	0	0	2	2
Numenta	0	0	0	1	0	0	1
SINA	0	0	0	0	1	0	1
Total from all contributors	0	0	528	12.033	14.007	5.046	31.614

Figure 6: TensorFlow - patches by authors affiliation (2018 covers January to March 26)

Long tail of “Unknown” contributors

The project has a long tail of hundreds of unaffiliated individual contributors sending one or two small patches using their private email address. This is a healthy sign that participants have enough interest and willingness to jump into an advanced topic, learn about it, and contribute. The community manager(s) of such a project would be ecstatic to have the chance work with such a community and harness the power of their contributors.

Microsoft Cognitive Toolkit (CNTK)

Project Creator	Microsoft
Description	The Microsoft Cognitive Toolkit (CNTK) is a toolkit for commercial-grade, distributed deep learning. It describes neural networks as a series of computational steps via a directed graph. CNTK allows the user to easily realize and combine popular model types such as feed-forward DNNs, convolutional neural networks (CNNs) and recurrent neural networks (RNNs/LSTMs). CNTK implements stochastic gradient descent (SGD, error backpropagation) learning with automatic differentiation and parallelization across multiple GPUs and servers.
Project History	Microsoft Research originally created the project and later released it under the MIT open source license on January 25, 2016.
Current Status	Active development. Last release was version 2.5 on March 15, 2018. Release notes are available on GitHub.
Releases	The project has gone through 35 releases since being on GitHub. The project maintains good releases documentation in addition to and specific instructions for each supported platform.
Roadmap	The project does not maintain a formal roadmap that captures target features and functionalities for future releases. The project had an effort to maintain a roadmap via "Iteration Plans" published on the GitHub wiki, however, it does not look that it is not up-to-date. However, we also noticed that the project captures some of its planning in GitHub issues.
License	MIT
Web Site	https://www.microsoft.com/en-us/cognitive-toolkit/

Blog	https://www.microsoft.com/en-us/cognitive-toolkit/blog/
Code Repository	https://GitHub.com/Microsoft/CNTK
Twitter	@MSCNTK
Platforms	Linux, Windows
APIs	CNTK provides libraries in Python, C++ for network composition and training, as well as for model evaluation. It also provides libraries in C#/.NET and Java to access the CNTK model evaluation facilities. Details on the available APIs are available from https://docs.microsoft.com/en-us/cognitive-toolkit/CNTK-Library-API .

Language Breakdown

































Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage	
 C++	202.569	36.181	15.2%	38.167	276.917		48.9%
 XML	129.733	356	0.3%	199	130.288		23.0%
 Python	71.109	25.595	26.5%	19.278	115.982		20.5%
 CUDA	9.928	1.302	11.6%	1.818	13.048		2.3%
 shell script	6.650	1.557	19.0%	2.204	10.411		1.8%
 C#	6.455	3.261	33.6%	1.328	11.044		2.0%
 Make	1.605	227	12.4%	421	2.253		0.4%
 Autoconf	1.419	0	0.0%	123	1.542		0.3%
 DOS batch script	1.267	71	5.3%	226	1.564		0.3%
 Perl	450	123	21.5%	66	639		0.1%
 HTML	443	0	0.0%	74	517		0.1%
 C	327	388	54.3%	194	909		0.2%
 Matlab	243	88	26.6%	40	371		0.1%
 Java	219	57	20.7%	37	313		0.1%
 XAML	138	0	0.0%	14	152		0.0%
 AWK	15	5	25.0%	4	24		0.0%
Totals	432.570	69.211		64.193	565.974		

Figure 7: CNTK is written primarily in C++ (48.9%), XML (23%), and Python (20.5%)

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

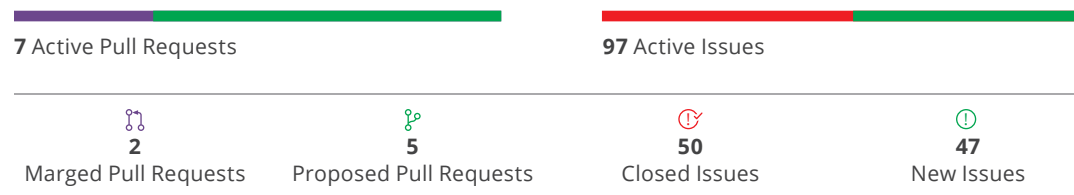


Figure 8: CNTK - GitHub project pulse

GitHub Project Insights

Figure 8 displays the GitHub project pulse for 1 month, March 15 to April 15, 2018. Excluding merges, 21 authors have pushed 28 commits to master and 173 commits to all branches. On master, 104 files have changed and there have been 5,455 additions and 2,074 deletions. Finally, 50 issues were closed and 47 new issues emerged.

Observations

Large and active development team

Figure 9 displays the number of unique contributors to the project listed by affiliations covering the period from January 1, 2014 until April 15, 2018.

Since it became an open source project, CNTK has seen code contributions from 257 unique developers, 114 of which are Microsoft employees. These developers have committed over 15,600 patches since CNTK became available as an open source project.

	2013	2014	2015	2016	2017	2018	Total
Microsoft	0	12	26	61	62	19	114
(Unknown)	0	6	11	47	54	17	120
(Academic)	0	0	4	4	1	0	8
NVIDIA	0	0	0	2	3	0	3
Mitsubishi Electric Research Laboratories	0	0	1	0	0	0	1
(Hobbyist)	0	0	0	3	1	0	4
Xevo	0	0	0	1	0	0	1
Yahoo!	0	0	0	1	3	0	4
Gitter	0	0	0	0	1	0	1
SINA	0	0	0	0	1	0	1
Total from all contributors	0	18	42	119	126	36	257

Figure 9: CNTK - Unique contributors by affiliation (2018 covers January to April 15)

Decreasing Y-O-Y development activity

Figure 10 presents the development activities in terms of commits to the project. Over the last 12 months, CNTK has seen a substantial decrease in development activity. There are fewer developers working on the project, which has a direct impact on the code contributions (added, removed LoC). This could mean many things. It may be a warning sign that interest in this project is waning, or it may indicate a maturing code base that requires fewer changes. We need to track the project a few more months in 2018 before we can identify a development trend with any certainty.

	2013	2014	2015	2016	2017	2018	Total
Microsoft	0	249	2,317	7,638	3,836	251	14,291
(Unknown)	0	31	430	195	384	47	1,087
(Academic)	0	0	156	17	10	0	183
NVIDIA	0	0	0	3	16	0	19
Mitsubishi Electric Research Laboratories	0	0	7	0	0	0	7
(Hobbyist)	0	0	0	8	7	0	15
Xevo	0	0	0	2	0	0	2
Yahoo!	0	0	0	1	6	0	7
Gitter	0	0	0	0	1	0	1
SINA	0	0	0	0	1	0	1
Total from all contributors	0	280	2,910	7,864	4,261	298	15,613

Figure 10: CNTK - patches by authors, by affiliation (2018 covers January to April 15)

Minimal to absent enterprise contributors

Only six companies contributed to the project outside of Microsoft. The combined contributions of these six companies are 37 patches. There are no external corporate contributions to the project in 2018 (up to the writing of this paper). This is alarming, especially when you look at other projects that were and are able to attract a much higher number of corporate contributors.

Fast release cycle

The table below displays all of the CNTK releases (including RCs and betas) since January 2017.

CNTK Release Version	Release Date
2.5	March 16, 2018
2.4	January 31, 2018
2.3	November 22, 2017
2.2	September 15, 2017
2.1	July 31, 2017
2.0	June 1, 2017
2.0 RC 3	May 24, 2017
2.0 RC 2	April 21, 2017

2.0 RC 1	April 3, 2017
2.0 Beta 15	March 15, 2017
2.0 Beta 12	February 23, 2017
2.0 Beta 11	February 10, 2017
2.0 Beta 11	February 1, 2017
2.0 Beta 9	January 20, 2017
CNTK 2.0 Beta 8	January 16, 2017

The project maintains a fast-paced development and release cycle. This is a very positive sign for active development, new features coming in, stabilization of code base, etc.

Convolutional Architecture for Fast Feature Embedding (Caffe)

Project Creator	The original creator of Caffe is Yangqing Jia during his PhD studies at the University of California at Berkeley. Yangqing is currently the Director of Facebook's AI infrastructure. The current Caffe lead developer from UC Berkeley is Evan Shelhamer.
Description	Caffe is a deep learning framework that supports many different types of deep learning architectures. The project is geared towards image classification and segmentation.
Current Status	Active development. Last release was version 1.0 on April 18, 2017 with only 96 commits to master since that release.
Releases	The project has gone through 14 releases since it was hosted on GitHub. Releases are well documented with direct links to download the source code in a *.zip format.
License	BSD 2-Clause License
Web Site	http://caffe.berkeleyvision.org/
Code Repository	https://GitHub.com/BVLC/caffe
Platforms	Linux, Windows, macOS
APIs	Caffe has command line, Python, and MATLAB interfaces. You can learn more by visiting http://caffe.berkeleyvision.org/tutorial/interfaces.html .

Language Breakdown












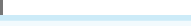







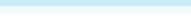


Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 C++	61.409	12.166	16.5%	8.182	81.757	 78.8%
 Python	5.448	3.036	35.8%	1.464	9.948	 9.6%
 CUDA	4.970	320	6.0%	641	5.931	 5.7%
 CMake	2.014	500	19.9%	447	2.961	 2.9%
 Matlab	627	205	24.6%	94	926	 0.9%
 shell script	551	197	26.3%	166	914	 0.9%
 Make	500	107	17.6%	92	699	 0.7%
 CSS	359	8	2.2%	71	438	 0.4%
 HTML	161	8	4.7%	19	188	 0.2%
 JavaScript	12	0	0.0%	0	12	 0.0%
 C	4	5	55.6%	3	12	 0.0%
Totals	76.055	16.552		11.179	103.786	

Figure 11: Caffe is written primarily in C++ (78.8%) with a small amount in Python (9.6%)

February 20, 2018 - March 20, 2018

Period: 1 month

Overview

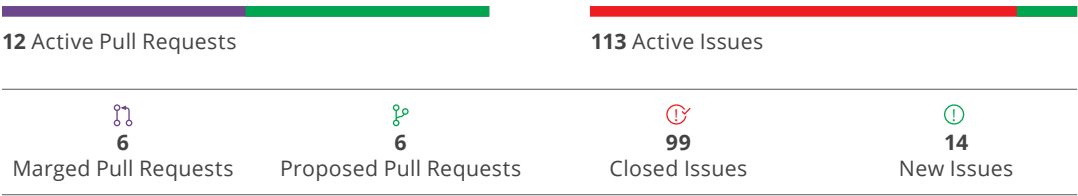


Figure 12: Caffè - GitHub project pulse for February 20 to March 20, 2018

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

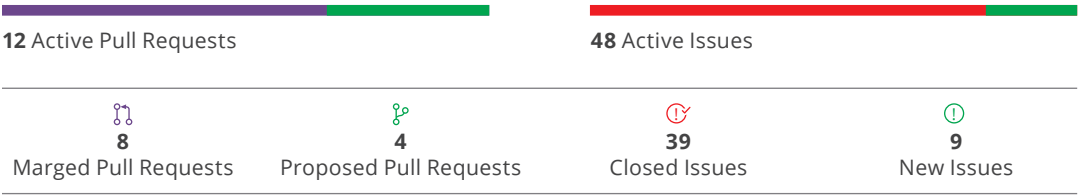


Figure 13: Caffè - GitHub project pulse for March 15 to April 15, 2018

GitHub Project Insights

Figure 12 displays the GitHub project pulse for 1 month, February 20 to March 20, 2018. Excluding merges, 4 authors have pushed 6 commits to master and 7 commits to all branches. On master, 15 files have changed and there have been 795 additions and 165 deletions. Finally, 99 issues were closed and 14 new issues emerged.

We came back to the project three weeks after we took that initial GitHub insight screenshot to see if there is the progress month over month. Excluding merges, 7 authors have pushed 6 commits to master and 11 commits to all branches. On master, 10 files have changed and there have been 398 additions and 23 deletions. Finally, 39 issues were closed and 9 new issues emerged.

Observations

Decreasing Y-O-Y development activity

As of 2015, the project has seen a decrease in development activity at all levels. There are fewer unique contributors, fewer patches getting into the project, with fewer changes to the code. This slowdown is associated with a number of factors; one of them is the move of some of the contributors to other projects. The biggest factor is the move into Caffe2 once Caffe 1.0 has been release. Below, we provide the note from the Caffe Team that was part of the release notes of Caffe 1.0 the transition to Caffe2.

Now that 1.0 is done, the next generation of the framework—Caffe2—is ready to keep up the progress on DIY deep learning in research and industry. While Caffe 1.0 development will continue with 1.1, Caffe2 is the new framework line for future development led by Yangqing Jia. Although Caffe2 is a departure from the development line of Caffe 1.0, we are planning a migration path for models just as we have future-proofed Caffe models in the past.

Figure 14 presents the committed patches per authors' affiliation.

	2013	2014	2015	2016	2017	2018	Total
(Academic)	324	1,539	612	217	103	6	2,801
(Unknown)	0	508	293	125	95	36	1,057
Senseta	0	0	3	0	0	0	3
(Hobbyist)	0	13	14	10	6	2	45
Google	9	94	7	0	0	0	110
Samsung	0	14	2	0	1	0	17
NVIDIA	0	0	28	6	1	0	35
Magisto	0	0	0	3	2	0	5
Zalando	0	0	0	2	0	0	2
Yahoo!	0	12	6	5	0	0	23
Intel	0	0	1	5	0	0	6

Pinterest	0	0	0	0	2	0	2
ACM	0	2	0	0	0	0	2
Max-Planck-Institut for Astrophysics	0	1	0	1	0	0	2
INRIA	0	0	0	2	0	0	2
MVTec Software GmbH	0	0	0	0	2	0	2
Adobe	0	0	1	0	0	0	1
Tencent Technology	0	0	1	0	0	0	1
Universal Audio	0	2	0	0	0	0	2
Baidu	0	0	0	1	0	0	1
IBM	0	0	0	2	0	0	2
SciLifeLab	0	0	1	0	0	0	1
Skydio	0	0	0	0	1	0	1
Mozilla	0	0	1	0	0	0	1
Alibaba	0	0	0	0	1	0	1
Qualcomm	0	0	0	0	1	0	1
Total from all contributors	333	2.185	970	379	215	44	4.126

Figure 14: Caffe – patches by authors, by affiliation (2018 covers January to April 15)

Decreasing enterprise contributors

Figure 15 illustrates the number of unique contributors per affiliations; the number of contributing organizations peaked in 2015 and has been declining since. November 2017 saw only one commit from a corporate contributor. Since then, there have been no enterprise contributors, i.e. contributors sending in patches of behalf of their companies.

	2013	2014	2015	2016	2017	2018	Total
(Academic)	5	15	18	14	8	1	37
(Unknown)	0	56	90	65	36	12	226
Senseta	0	0	1	0	0	0	1
(Hobbyist)	0	3	6	4	2	1	13
Google	1	2	2	0	0	0	3
Samsung	0	1	1	0	1	0	3
NVIDIA	0	0	5	2	1	0	6
Magisto	0	0	0	1	1	0	1
Zalando	0	0	0	1	0	0	1
Yahoo!	0	3	2	3	0	0	7
Intel	0	0	1	5	0	0	5
Pinterest	0	0	0	0	1	0	1
ACM	0	1	0	0	0	0	1
Max-Planck-Institut for Astrophysics	0	1	0	1	0	0	2
INRIA	0	0	0	2	0	0	2
MVTec Software GmbH	0	0	0	0	1	0	1
Adobe	0	0	1	0	0	0	1
Tencent Technology	0	0	1	0	0	0	1
Baidu	0	0	0	1	0	0	1
Universal Audio	0	1	0	0	0	0	1

Figure 15: Caffe - Unique contributors, by affiliation

IBM	0	0	0	2	0	0	2
SciLifeLab	0	0	1	0	0	0	1
Skydio	0	0	0	0	1	0	1
Alibaba	0	0	0	0	1	0	1
Mozilla	0	0	1	0	0	0	1
Qualcomm	0	0	0	0	1	0	1
Total from all contributors	6	83	130	101	54	14	320

Figure 15: Caffe - Unique contributors, by affiliation

Competing open source AI projects

Caffe was a pioneer in open source AI being one of the early projects. Since then, the open source AI ecosystem has grown so much that it now includes dozens of competing open source AI frameworks and libraries. Some companies such as Google, Facebook, Microsoft, IBM and Apple also had their own internal efforts that were released as open source, and they lobbied their partners to join these efforts. As a result, it is an open buffet for companies who are yet undecided with plenty of choices depending on the problems they want to target.

Serving an academic purpose

Caffe continues to be project with an academic purpose supported by the Berkeley Vision and Learning Center and helping push AI rich academics into the commercial world.

Caffe2

Project Creator	Facebook
Description	Caffe2 is a deep learning framework enabling simple and flexible deep learning. Built on the original Caffe, Caffe2 is designed with expression, speed, and modularity in mind, allowing a more flexible way to organize computation.
Caffe vs. Caffe2	<p>The original Caffe framework was useful for large-scale product use cases, especially with its unparalleled performance and well tested C++ codebase. Caffe2 has some design choices that are inherited from its original use case: conventional CNN applications. As new computation patterns have emerged, especially distributed computation, mobile, reduced precision computation, and more non-vision use cases, its design has shown some limitations.</p> <p>Caffe2 improves Caffe 1.0 in a number of areas and directions:</p> <ul style="list-style-type: none"> • Support for large-scale distributed training. • Mobile deployment. • New hardware support (in addition to CPU and CUDA). • Flexibility for future directions such as quantized computation. • Stress tested by the vast scale of Facebook applications.
Current Status	<p>Active development.</p> <p>Last release was version 0.8.1 on August 8, 2017. There has been 1385 commit to the master branch since then. For release history, please visit https://GitHub.com/caffe2/caffe2/releases.</p>
Releases	The project has gone through 4 releases since it was hosted on GitHub. The project offers release notes and installation instructions.
License	MIT

Web Site	https://caffe2.ai
Blog	https://caffe2.ai/blog/
Code Repository	https://GitHub.com/caffe2/caffe2
Twitter	@Caffe2ai
Platforms	Linux, Windows, macOS, CentOS, iOS, Android, Raspbian, Tegra
APIs	Caffe2 comes with native Python and C++ APIs that work interchangeably so you can prototype quickly and easily optimize later. To learn more about the APIs, please visit https://caffe2.ai/docs/api-intro.html .

Language Breakdown






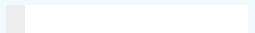

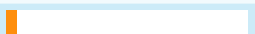



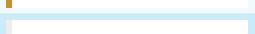












Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 C++	129.587	25.375	16.4%	18.295	173.257	 53.5%
 Python	64.604	15.418	19.3%	12.711	92.733	 28.6%
 CUDA	19.821	2.542	11.4%	2.368	24.731	 7.6%
 C	10.734	1.181	9.9%	1.630	13.545	 4.2%
 Objective-C	6.231	378	5.7%	718	7.327	 2.3%
 CMake	4.876	1.956	29.6%	908	7.740	 2.4%
 CSS	1.355	82	5.7%	284	1.721	 0.5%
 shell script	1.146	402	26.0%	258	1.806	 0.6%
 XML	370	14	3.6%	10	394	 0.1%
 HTML	205	33	13.9%	8	246	 0.1%
 DOS batch script	126	26	17.1%	31	183	 0.1%
 Make	36	5	12.2%	18	59	 0.0%
Totals	239.091	47.412		37.239	323.742	

Figure 16 - Caffe 2 is written primarily in C++ (53.5%) and Python (28.8%)

February 20, 2018 - March 20, 2018

Period: 1 month

Overview

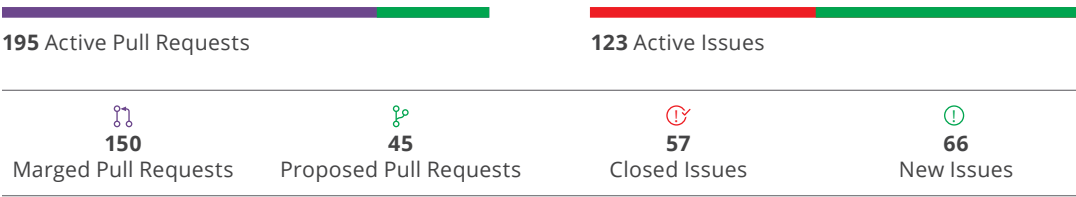


Figure 17: Caffè2 - GitHub project pulse from February 20 to March 20, 2018

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

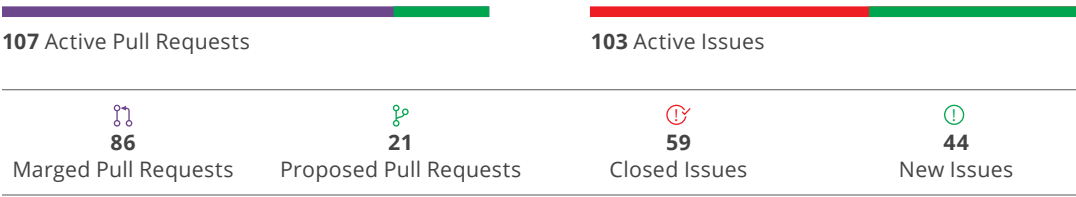


Figure 18: Caffè2 - GitHub project pulse from March 15 to April 15, 2018

GitHub Project Insights

Figure 17 displays the GitHub project pulse for 1 month, February 20 to March 20, 2018. Excluding merges, 51 authors have pushed 252 commits to master and 304 commits to all branches. On master, 523 files have changed and there have been 55,976 additions and 4,266 deletions. Finally, 57 issues were closed and 6 new issues emerged.

We came back to the project three weeks after we took that initial GitHub insight screenshot to see if there is the progress month over month (Figure 18). Excluding merges, 50 authors have pushed 171 commits to master and 374 commits to all branches. On master, 1,566 files have changed and there have been 12,327 additions and 37,663 deletions. Finally, 59 issues were closed and 44 new issues emerged.

Observations

Increasing Y-O-Y development activity

The development activity on the project continues on an upward trend since its creation both in terms of commits pouring into the project and the number of added and/or removed LoC. Figure 19 illustrates the number of commits accepted in the project since 2015. The development activities are on an upward trend with Facebook doing most of the heavy lifting.

	2013	2014	2015	2016	2017	2018	Total
Facebook	0	0	7	285	2,184	345	2.821
Google	0	0	170	24	0	0	203
(Unknown)	0	0	1	51	264	238	554
(Academic)	0	0	12	1	17	32	62
Intel	0	0	0	0	2	0	2
Instagram	0	0	0	7	5	1	13
NVIDIA	0	0	0	0	11	0	11
Samsung	0	0	0	0	1	2	3
Yahoo!	0	0	0	0	2	1	3
Unifie	0	0	0	0	1	0	1
eTuning Foundation	0	0	0	0	1	0	1
Baidu	0	0	0	0	1	0	1
(Hobbyist)	0	0	0	0	2	0	2
Alibaba Cloud	0	0	0	0	1	0	1
Total from all contributors	0	0	199	368	2.402	619	3.678

Figure 19: Caffe2 - Patches by authors, by affiliation (2018 covers January to March 25)

Lack of enterprise contributors

Figure 20 illustrates the unique contributors to Caffe2 by affiliation. Outside of Facebook and Instagram, there are very few contributions and participation from other companies. Corporate contributors have contributed less than 25 patches to Caffe2 since its debut in open source with NVidia being the top corporate contributor with 11 patches (+309, -273). In 2017, the project had 281 unique contributors compared to 110 so far in 2018. This is not particularly alarming since the unique contributors on a monthly basis in 2018 do actually match those of 2017. The number of unique contributors of the remaining of 2018 will tell if this trend is moving in a specific direction.

	2013	2014	2015	2016	2017	2018	Total
Facebook	0	0	1	33	203	79	243
Google	0	0	1	1	0	0	1
(Unknown)	0	0	1	5	60	37	96
(Academic)	0	0	2	1	5	7	14
Intel	0	0	0	0	2	0	2
Instagram	0	0	0	1	2	1	3
NVidia	0	0	0	0	2	0	2
Samsung	0	0	0	0	1	1	1
Yahoo!	0	0	0	0	1	1	1
Unifie	0	0	0	0	1	0	1
eTuning Foundation	0	0	0	0	1	0	1
(Hobbyist)	0	0	0	0	1	0	1
Baidu	0	0	0	0	1	0	1
Alibaba Cloud	0	0	0	0	1	0	1
Total from all contributors	0	0	5	40	281	126	367

Figure 20: Caffe2 - Unique contributors, by affiliation (2018 covers January to April 15)

Theano

Project Creator	Montreal Institute for Learning Algorithms (https://mila.quebec) at the Université de Montréal.
Description	Theano is a Python library that lets you to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays.
Current Status	<p>Maintenance mode only. Major development stopped after the 1.0.0 release.</p> <p>Last release was version 1.0.0 on November 15, 2017.</p> <p>On September 28, 2017, Pascal Lamblin posted a message from Yoshua Bengio (Head of MILA) that major development would cease after the 1.0 release due to competing offerings by strong industrial players. Here is the link to the actual announcement.</p>
License	BSD 3-Clause
Web Site	http://deeplearning.net/software/theano/
Code Repository	https://GitHub.com/Theano/
Platforms	Linux, Windows
APIs	Thenao provides Python APIs. Documentation on this aspect is available from http://deeplearning.net/software/theano/library/index.html .

Contribution History

Figure 21, a screenshot from our Facade analysis, illustrates the commit history of the project from January 1, 2013 to March 20, 2018. You can notice the severe contribution drop in 2018 due to halting active development. The same applies to the number of unique contributors to the project as shown in Figure 22.

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	637	1.457	2.735	2.028	2.283	36	9.176
(Hobbyist)	1.153	1.144	929	764	581	16	4.587
(Academic)	412	254	794	501	430	8	2.399
NVidia	0	0	6	0	14	0	20
Yahoo!	0	0	0	0	14	0	14
Intel	0	0	0	7	1	0	8
Zoho Cloud Software	0	0	0	0	5	0	5
Plan Grid	0	0	2	0	0	0	2
Google	0	2	0	2	0	0	4
CliniComp Intl.	0	0	0	0	4	0	4
X.AI	0	0	0	0	7	0	7
CERN	0	0	0	2	0	0	2
Huawei	0	0	0	0	1	0	1
DeepMind	0	0	0	0	1	0	1
Adobe	0	1	0	0	0	0	1
Total from all contributors	2.202	2.858	4.466	3.304	3.341	60	16.231

Figure 21: Theano - patches by author affiliation (2018 covers January to March 20)

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	44	61	114	124	83	8	322
(Hobbyist)	3	4	7	6	5	1	13
(Academic)	5	18	32	29	27	2	87
NVidia	0	0	1	0	1	0	2
Yahoo!	0	0	0	0	1	0	1
Intel	0	0	0	1	1	0	2
Zoho Cloud Software	0	0	0	0	2	0	2
Plan Grid	0	0	1	0	0	0	1
Google	0	2	0	1	0	0	3
CliniComp Intl.	0	0	0	0	1	0	1
X.AI	0	0	0	0	1	0	1
CERN	0	0	0	2	0	0	3
Huawei	0	0	0	0	1	0	1
DeepMind	0	0	0	0	1	0	1
Adobe	0	1	0	0	0	0	1
Total from all contributors	52	86	155	163	123	11	439

Figure 22: Theano – unique contributors by affiliation (2018 covers January 1 to March 20)

Language Breakdown























Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Python	129.239	54.628	29.7%	35.142	219.009	 86.6%
 C	18.984	5.091	21.1%	1.916	25.991	 10.3%
 Tex/LaTex	2.616	286	9.9%	415	3.317	 1.3%
 JavaScript	1.932	339	14.9%	274	2.545	 1.0%
 shell script	550	158	22.3%	172	880	 0.3%
 CUDA	292	100	25.5%	59	451	 0.2%
 CSS	224	1	0.4%	40	265	 0.1%
 HTML	141	1	0.7%	8	150	 0.1%
 DOS batch script	103	26	20.2%	33	162	 0.1%
 Make	23	3	11.5%	9	36	 0.0%
 C++	2	16	88.9%	3	21	 0.0%
Totals	154.106	60.649		38.071	252.826	

Figure 23: Theano is primarily written in Python (86.6%)

February 21, 2018 - March 21, 2018

Period: 1 month

Overview

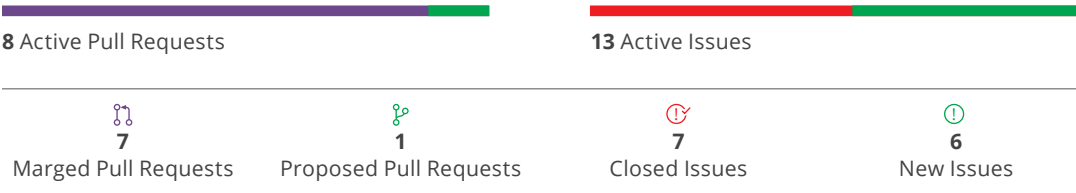


Figure 24: Theano - GitHub project pulse covering February 21 to March 21, 2018

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

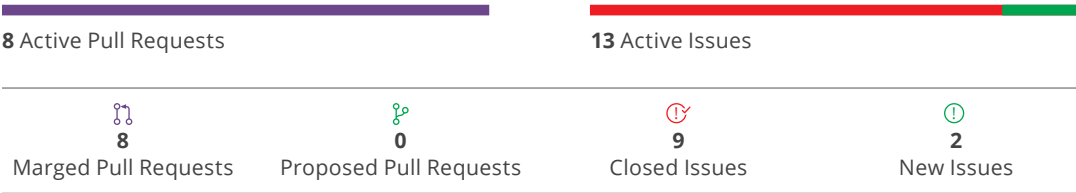


Figure 25: Theano - GitHub project pulse covering March 15 to April 15, 2018

GitHub Project Insights

Figure 24 displays the project pulse in GitHub for 1 month, February 21 to March 21, 2018. Even though the project has official ceased active development, there is still a certain minimal level of ongoing activities. Excluding merges, 3 authors have pushed 21 commits to master and 21 commits to all branches. On master, 20 files have changed and there have been 318 additions and 85 deletions. Finally, 7 issues were closed and 6 new issues emerged.

We came back to the project three weeks after taking the initial GitHub insight screenshot to see if there is was progress month over month (Figure 25).

Excluding merges, 3 authors have pushed 21 commits to master and 21 commits to all branches. On master, 22 files have changed and there have been 296 additions and 55 deletions. Finally, 9 issues were closed and 2 new issues emerged.

Observations

Due to halting active development on the project after the 1.0.0 release, the project has suffered a major blow in terms of the number of contributor and the amount of contributions getting into it. Furthermore, all corporate developers have left the project and the only active developers now are either hobbyist or from academia.

We believe there is great value in Theano's code base because it is relatively mature since it was one of the early open source AI projects with the first lines of code added in early 2008, way before many of the projects listed in this report even existed. Such a lengthy source control history and the ability of the project to attract a significant number of contributors from the academia and the industry is a good indication of the value that resides in the project. We hope that the project continues to exist as a training ground for students and academic especially given its roots at the Université de Montréal.

Torch

Project Creator	<p>The original authors of Torch are:</p> <ul style="list-style-type: none"> • Ronan Collobert (now, Research Scientist at Facebook) • Koray Kavukcuoglu (now, Research Scientist with Google) • Clement Farabet (now, VP of AI Infrastructure at NVidia)
Description	Torch is a machine-learning library, a scientific computing framework, and a script language based on the Lua programming language. It provides a wide range of algorithms for deep machine learning.
Current Status	Latest stable release was version 7 on February 27, 2017. The project is now in maintenance mode with only bug fixes being committed.
Roadmap	The project maintains a roadmap document in the GitHub repository.
License	BSD 3-Clause
Web Site	http://torch.ch
Blog	http://torch.ch/blog/index.html
Code Repository	https://GitHub.com/torch/torch7
Twitter	@TorchML
Platforms	Linux, Android, macOS, iOS, Windows
APIs	Lua, C, C++/OpenCL

Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C	19.605	1.221	5.9%	3.147	23.973	66.1%
Lua	8.637	493	5.4%	1.229	10.359	28.6%
CMake	1.471	186	11.2%	207	1.864	5.1%
C++	50	16	24.2%	18	84	0.2%
Totals	29.763	1.916		4.601	36.280	

Figure 26: Torch is written primarily in C (66.1%) and Lua (28.6%)

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

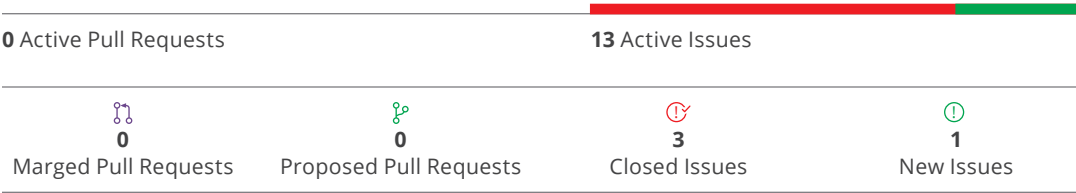


Figure 27: Torch - GitHub project pulse for March 15 to April 15, 2018

GitHub Project Insights

Figure 27 displays the project pulse in GitHub for 1 month, March 15 to April 15, 2018, illustrating no development on the project.

Contribution History

Figure 28 illustrates the commit history of the project covering the period from January 1, 2013 to April 15, 2018. You can notice the drop in contributions in 2018. The same applies to the number of unique contributors as shown in Figure 29.

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	71	155	272	198	136	0	832
Idiap Reseach	11	43	0	0	0	0	54
Google	0	10	58	20	0	0	88
Facebook	0	16	37	26	42	0	121
Twitter	0	2	19	7	11	0	39
(Hobbyist)	3	6	3	4	2	0	18
(Academic)	0	0	4	6	8	0	18
Mobile Vision	0	1	11	4	0	0	16
Xamla Robotic Solutions	0	0	0	2	0	0	2
NEC Labs America	0	0	0	1	2	0	3
Orobix	0	0	0	0	2	0	2
Namvii World	0	0	2	0	0	0	2
NVidia	0	0	1	3	1	0	5
Microsoft	0	0	0	1	0	0	1
Instituto Eldorado	0	0	0	0	2	0	2
Gitter	0	0	1	0	0	0	1
Total from all contributors	85	233	408	272	206	0	1.204

Figure 28: Torch - patches by authors, by affiliation (2018 covers January 1 to April 15)

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	7	14	43	34	25	0	93
Idiap Reseach	1	1	0	0	0	0	1
Google	0	5	14	11	0	0	23
Facebook	0	3	6	5	8	0	13
Twitter	0	1	2	4	3	0	6
(Hobbyist)	1	2	3	1	1	0	6
(Academic)	0	0	1	5	4	0	8
Mobile Vision	0	1	2	1	0	0	2
Xamla Robotic Solutions	0	0	0	1	0	0	1
NEC Labs America	0	0	0	1	1	0	1
Orobix	0	0	0	0	1	0	1
Namvii World	0	0	1	0	0	0	1
NVidia	0	0	1	1	1	0	2
Microsoft	0	0	0	1	0	0	1
Instituto Eldorado	0	0	0	0	1	0	1
Gitter	0	0	1	0	0	0	1
Total from all contributors	9	26	74	65	45	0	160

Figure 29: Torch – unique contributors by affiliation

Accord.NET

Project Creator	The original author was César Roberto de Souza.
Description	The Accord.NET Framework is a machine learning framework combined with audio and image-processing libraries for building production-grade computer vision, computer audition, signal processing and statistics applications.
Current Status	<p>It appears that development has ceased with zero contributions or commits in 2018.</p> <p>Last release was version 3.8.0 on October 22, 2017. Release history and notes are available from https://GitHub.com/accord-net/framework/releases.</p>
Releases	The project has gone through 15 releases since it was hosted on GitHub. The project maintains release notes and provides links to the commits that were accepted into the master branch.
License	LGPL 2.1
Web Site	http://accord-framework.net/
Code Repository	https://GitHub.com/accord-net/framework

Language Breakdown










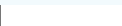





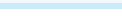

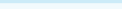





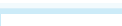


Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 XML	1,427,259	38,288	2.6%	150,953	1,616,500	 50.7%
 C#	833,399	455,479	35.3%	181,034	1,469,912	 46.1%
 C++	17,556	26,890	60.5%	5,813	50,259	 1.6%
 C	13,183	26,562	66.8%	5,032	44,777	 1.4%
 XAML	1,932	24	1.2%	16	1,972	 0.1%
 XML Schema	628	8	1.3%	82	718	 0.0%
 F#	428	114	21.0%	175	717	 0.0%
 DOS batch script	284	117	29.2%	33	434	 0.0%
 Visual Basic	283	144	33.7%	109	536	 0.0%
 HTML	272	6	0.0%	6	278	 0.0%
 shell script	142	7	4.7%	33	182	 0.0%
 Automake	88	13	12.9%	24	125	 0.0%
 Autoconf	30	2	6.3%	14	46	 0.0%
Totals	2,295,484	547,648		343,326	3,186,456	

Figure 30: Accord.NET is written primarily in XML (50.7%) and C# (46.1%)

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

1 Active Pull Requests

13 Active Issues





 0	 1	 11	 44
Marged Pull Requests	Proposed Pull Requests	Closed Issues	New Issues

Figure 31: Accord.NET – GitHub project pulse for March 15 to April 15, 2018.

GitHub Project Insights

Figure 31 displays the project pulse in GitHub for 1 month, March 15 to April 15, 2018.

Contribution History

Figure 32 is a screenshot from our Facade analysis that illustrates the commit history of the project from January 1, 2013 to March 20, 2018. You can notice the halt in contributions in 2018.

	2013	2014	2015	2016	2017	2018	Total
Xerox	70	289	290	479	412	0	1.540
Daitan Group	31	0	0	0	0	0	31
Naver Labs Europe	0	0	0	0	558	0	558
(Hobbyist)	0	0	0	1	3	0	4
Cicso	0	3	1	0	0	0	4
Cureos	0	27	12	2	0	0	41
(Unknown)	1	18	41	14	133	0	207
Mcmaster-Carr	0	0	24	0	0	0	24
Freebox Revolution	0	0	1	0	0	0	1
ABC Arbitrage	0	0	1	1	1	0	3
DanEst Consulting	0	0	0	0	3	0	3
PARASCRIPT	0	0	0	0	2	0	2
Microsft	0	0	1	2	0	0	3
AnatemetA Cloud Services	0	0	0	1	0	0	1
ACM	0	2	0	0	0	0	2
Weingartner Maschinenbau GmbH	0	0	0	0	2	0	2
DriveTime	0	0	0	1	0	0	1
Shift Technology	0	1	0	0	0	0	1
e-FRACIAL	0	0	0	1	0	0	1
Catalysts	0	1	0	0	0	0	1
Gitter	0	0	1	0	0	0	1
Collibris	0	0	0	0	1	0	1
Total from all contributors	102	341	372	502	1.115	0	2.432

Figure 32: Accord.NET – patches by author affiliation (2018 covers January to March 20)

Figure 33 highlights the unique contributors the project grouped by their affiliation.

	2013	2014	2015	2016	2017	2018	Total
Xerox	1	1	1	1	1	0	1
Daitan Group	1	0	0	0	0	0	1
Naver Labs Europe	0	0	0	0	1	0	1
(Hobbyist)	0	0	0	1	1	0	2
Cicso	0	1	1	0	0	0	1
Cureos	0	1	1	1	0	0	1
(Unknown)	1	6	10	9	22	0	46
Mcmaster-Carr	0	0	1	0	0	0	1
Freebox Revolution	0	0	1	0	0	0	1
ABC Arbitrage	0	0	1	1	1	0	3
DanEst Consulting	0	0	0	0	1	0	1
PARASCRIP	0	0	0	0	1	0	1
Microsft	0	0	1	1	0	0	2
AnatemetA Cloud Services	0	0	0	1	0	0	1
ACM	0	1	0	0	0	0	1
Weingartner Maschinenbau GmbH	0	0	0	0	1	0	1
DriveTime	0	0	0	1	0	0	1
Shift Technology	0	1	0	0	0	0	1
e-FRACIAL	0	0	0	1	0	0	1
Catalysts	0	1	0	0	0	0	1
Gitter	0	0	1	0	0	0	1
Collibris	0	0	0	0	1	0	1
Total from all contributors	3	12	18	17	29	0	69

Figure 33: Accord.NET – unique contributors by affiliation

Apache SINGA

Project Host	Apache Software Foundation
Project History	<p>The DB System Group at the National University of Singapore initiated the project in 2014 in collaboration with the database group of Zhejiang University.</p> <p>The Apache Incubator accepted the project in March 17, 2015.</p>
Description	An Apache incubating project for developing an open source machine-learning library. It provides a flexible architecture for scalable distributed training, is extensible to run over a wide range of hardware, and has a focus on health-care applications. The project focuses on distributed deep learning by partitioning the model and data onto nodes in a cluster and parallelizing the training.
Current Status	<p>Active development.</p> <p>Last release was version 1.1.1 on June 29, 2017.</p>
Releases	The project has gone through 9 releases since it was hosted on GitHub. There are no release notes but the project provides links to the commits that were accepted as part of the release.
Roadmap	<p>The project maintains a development schedule with features targeted for each release.</p> <p>The roadmap is available at http://singa.apache.org/en/develop/schedule.html.</p>
License	Apache 2.0
Web Site	http://singa.apache.org/

Code Repository	https://GitHub.com/apache/incubator-singa
Twitter	@ApacheSinga
Platforms	Linux, macOS, Windows
APIs	Python, C++ and Java.

Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	37.832	12.866	25.4%	6.390	57.088	70.6%
Python	9.087	5.593	38.1%	2.459	17.139	21.2%
C	2.933	625	17.6%	336	3.894	4.6%
CMake	700	379	35.1%	164	1.243	1.5%
CUDA	404	191	32.1%	79	674	0.8%
shell script	193	280	59.2%	45	518	0.6%
HTML	91	4	4.2%	16	111	0.1%
Java	38	34	47.2%	17	89	0.1%
XML	37	17	31.5%	1	55	0.1%
CSS	3	0	0.0%	0	3	0.0%
Totals	51.318	19.989		9.507	80.814	

Figure 34: Apache SINGA is written primarily in C++ (70.6%) and Python (21.2%)

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

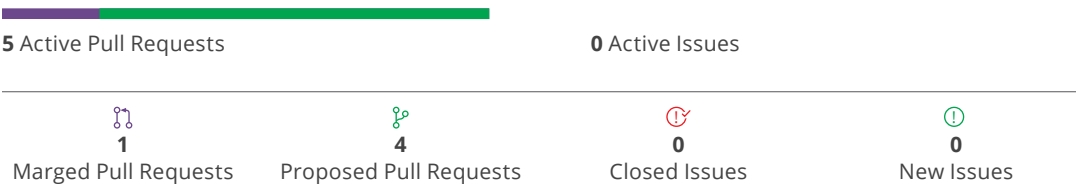


Figure 35: Apache SINGA - GitHub project pulse for March 15 to April 15, 2018

GitHub Project Insights

Figure 35 displays the project pulse in GitHub for 1 month, March 15 to April 15, 2018.

Observations

The project is currently in incubation at the Apache Software Foundation. The initial contributions were in April 2015, almost three years ago, and the project maintained a stable development state from 2015 to 2017 in terms of unique contributors. In 2017, the number of contributions started to drop and suffered a severe drop in 2018.

	2013	2014	2015	2016	2017	2018	Total
(Academic)	0	0	8	8	8	2	12
(Unknown)	0	0	8	14	12	2	25
(Hobbyist)	0	0	0	1	0	0	1
Tencent Technology	0	0	1	0	0	0	1
Total from all contributors	0	0	17	23	20	4	39

Figure 36: SINGA - unique authors by affiliation (2018 covers January to April 15)

Given the lack of corporate contributors who often boost a project significantly, it is hard to state that the project has passed its critical early start-up period, and has become established. Figure 36 presents the number of unique contributors to the project grouped by affiliation. Although the project's state can rapidly change, at the time of writing, SING looks more of an experiment that will remain in incubation. Figure 37 shows the number of patches grouped by affiliation since the project's inception.

	2013	2014	2015	2016	2017	2018	Total
(Academic)	0	0	213	240	94	5	552
(Unknown)	0	0	133	138	41	3	315
(Hobbyist)	0	0	0	1	0	0	1
Tencent Technology	0	0	1	0	0	0	1
Total from all contributors	0	0	347	379	135	8	869

Figure 37: SINGA - patches by author affiliation (2018 covers January to March 25)

Apache Mahout

Project Host	Apache Software Foundation
Description	The goal of the project is to build an environment for quickly creating scalable and performant machine learning applications. Apache Mahout offers a distributed linear algebra framework and mathematically expressive Scala DSL designed to let mathematicians, statisticians and data scientists quickly implement their own algorithms. The Apache Software Foundation recommends the Apache Spark as the out-of-the box distributed back-end. We cover Apache Spark in a later section.
Current Status	<p>It appears that development has ceased with zero code contributions in 2018. Starting with the release 0.10.0, the project shifts its focus to building backend-independent programming environment, code named Samsara.</p> <p>Last stable release was version 0.13.0 on April 14, 2017. Latest candidate release was 0.13.1-rc1 on July 6, 2017.</p>
Releases	The project has gone through 25 releases since it was hosted on GitHub. There are no release notes but the project provides links to the commits that were accepted as part of the release.
License	Apache 2.0
Web Site	https://mahout.apache.org/
Code Repository	https://GitHub.com/apache/mahout

Twitter	@ApacheMahout
APIs	Details on the available APIs are available from http://mahout.apache.org/docs/0.13.0/api/docs/ .

Languages Breakdown






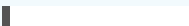



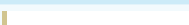

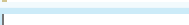










Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Java	110.928	48.201	30.3%	23.531	182.660	 77.1%
 Scala	15.952	8.679	35.2%	6.371	31.002	 13.1%
 XML	6.432	870	11.9%	473	7.766	 3.3%
 JavaScript	5.390	1.396	20.6%	1.588	8.374	 3.5%
 Perl	3.538	49	1.4%	455	4.042	 1.7%
 shell script	1.161	517	30.8%	261	1.939	 0.8%
 HTML	341	53	13.5%	82	476	 0.2%
 Python	248	54	17.9%	142	444	 0.2%
 R	90	23	20.4%	68	181	 0.1%
 XSL Transformation	29	16	35.6%	9	54	 0.0%
 Ruby	10	13	56.5%	5	28	 0.0%
Totals	144.110	59.871		32.985	236.966	

Figure 38: Apache Mahout is written primarily in Java (77.1%) and Scala (13.1%).

Contribution History

Figure 39 and Figure 40 from our Facade analysis illustrate respectively the commit history of the project from January 1, 2013 to April 15, 2018 and the number of unique contributors by affiliation. The project had a good run from 2013 to 2017 and mostly ceased development in 2018.

	2013	2014	2015	2016	2017	2018	Total
IBM	0	0	0	0	140	0	140
Apache Software Foundation	188	128	140	255	103	0	814
(Hobbyist)	218	126	14	8	2	0	368
Start Bootstrap	0	0	0	0	1	0	1
(Unknown)	0	13	57	3	6	0	79
Intel	0	4	50	0	0	0	54
LucidWorks	0	0	0	0	2	0	2
Red Hat	0	0	5	0	0	0	5
Cloudera	0	2	0	0	0	0	2
Accenture	0	2	0	0	0	0	2
Lawrence Livermore National Laboratory	0	0	2	1	0	0	3
(Academic)	0	1	0	0	0	0	1
Total from all contributors	406	276	268	267	254	0	1.471

Figure 39: Mahout - patches by author affiliation (2018 covers January to April 15)

	2013	2014	2015	2016	2017	2018	Total
IBM	0	0	0	0	2	0	2
Apache Software Foundation	10	6	4	4	4	0	15
(Hobbyist)	4	6	1	2	1	0	8
Start Bootstrap	0	0	0	0	1	0	1
(Unknown)	0	3	3	2	5	0	12
Intel	0	1	1	0	0	0	1
LucidWorks	0	0	0	0	1	0	1
Red Hat	0	0	1	0	0	0	1
Cloudera	0	1	0	0	0	0	1
Accenture	0	1	0	0	0	0	1
Lawrence Livermore National Laboratory	0	0	1	1	0	0	1
(Academic)	0	1	0	0	0	0	1
Total from all contributors	14	19	11	9	14	0	45

Figure 40: Mahout - unique contributors by affiliation (2018 covers January to April 15)

Apache Spark

Project Host	Apache Software Foundation
Project History	<p>The project was originally developed at the University of California, Berkeley's AMPLab by Matei Zaharia.</p> <p>In 2013, the project was donated to the Apache Software Foundation and switched its license from BSD to Apache 2.0.</p> <p>In February 2014, Spark became a top-level Apache project.</p> <p>It is worth mentioning that Matei Zaharia founded a company called Databricks to help clients with cloud-based big data processing using Spark. Databricks is the largest corporate contributor to Spark.</p>
Description	Apache Spark is an Open Source cluster-computing framework for fast and flexible large-scale data analysis. It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.
Current Status	<p>Active development.</p> <p>Last release was version 2.3.0 on February 28, 2018.</p>
Releases	The project has gone through 59 releases since it was hosted on GitHub. There are no release notes but the project provides links to the commits that were accepted as part of the release.
Roadmap	The project does not maintain a formal roadmap that captures target features and functionalities for future releases. The project had an effort to maintain a roadmap via "Iteration Plans" published via the GitHub wiki, however, it does not look that it is being put to use. We also noticed that some planning is actually captured in GitHub issues.

License	Apache 2.0
Web Site	https://spark.apache.org/
Code Repository	https://GitHub.com/apache/spark
Platforms	Linux, Windows, macOS
APIs	<ul style="list-style-type: none">• Spark Scala API (Scaladoc)• Spark Java API (Javadoc)• Spark Python API (Sphinx)• Spark R API (Roxygen2)• Spark SQL, Built-in Functions (MkDocs)

Languages Breakdown






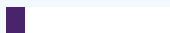

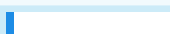







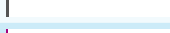







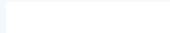



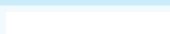
Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage	
 Scala	887.736	322.560	26.7%	172.260	1.382.556		68.5
 Java	217.669	57.947	21.0%	43.155	318.771		15.8
 Python	76.818	55.645	42.0%	25.740	158.203		7.8
 R	29.645	27.755	48.4%	6.549	63.949		3.2
 SQL	27.699	2.273	7.6%	2.521	32.493		1.6
 CSS	15.546	508	3.2%	2.346	18.400		0.9
 XML	13.900	2.613	15.8%	842	17.355		0.9
 JavaScript	9.344	2.536	21.3%	2.041	13.921		0.7
 shell script	4.650	3.499	42.9%	1.446	9.595		0.5
 HTML	1.002	86	7.9%	92	1.180		0.1
 DOS batch script	442	44	9.1%	78	564		0.0
 Ruby	388	118	23.3%	130	636		0.0
 Make	302	44	12.7%	66	412		0.0
 C	38	40	51.3%	20	98		0.0
Totals	1.285.179	475.668		257.286	2.018.133		

Figure 41: Spark is written primarily in Scala (68.5%) and Java (15.8%)

Observations

The Apache Spark project has almost a decade of history and source code contributions. The project is well established with a very strong community of contributors from academia and the industry. In fact, the number of contributors (by affiliation) is so large that it needs three A4 papers to fit it. Instead of providing that very long list, we offer a summary of the various metrics in the table below. Please note that the stats for 2018 cover January 1 until April 15.

Metric/Year	2013	2014	2015	2016	2017	2018
Added LoC	189,506	544,545	554,280	450,672	249,941	54,468
Removed LoC	145,375	172,258	253,463	228,036	105,146	30,141
Patches	3,914	3,569	5,030	4,342	2,526	626
Unique Contributors	153	410	661	482	426	152

Over the past twelve months, over 400 developers contributed new code to the project. Spark has one of the largest contributor populations we have seen among the projects we surveyed. In addition, the project has over 80 companies contributing to it making it the project with the strongest corporate contributions. It has very strong development activity, and in the period ranging from February 26 to March 26, excluding merges, 68 authors have pushed 131 commits to the master branch and 195 commits to all branches.

Eclipse DeepLearning4J (DL4J)

Project Creator	<p>Adam Gibson, Chris Nicholson, and Josh Patterson are the primary creators of DL4J. They are also co-founders of a startup called Skymind, which bundles DL4J, TensorFlow, Keras and other deep learning libraries in an enterprise distribution called the Skymind Intelligence Layer.</p> <p>DL4J was contributed to the Eclipse Foundation in October 2017.</p>
Description	<p>Eclipse Deeplearning4j is a deep learning programming library written for Java and the Java virtual machine, and a computing framework with wide support for deep learning algorithms. Deeplearning4j includes implementations of the restricted Boltzmann machine, deep belief net, deep autoencoder, stacked denoising autoencoder and recursive neural tensor network, word2vec, doc2vec, and GloVe. These algorithms all include distributed parallel versions that integrate with Apache Hadoop and Spark.</p>
Current Status	<p>Active development.</p> <p>Last release was version 0.9.1 on August 12, 2017.</p>
Releases	<p>The project has gone through 47 releases since it was hosted on GitHub. There are no release notes but the project provides links to the commits that were accepted as part of the release.</p>
Roadmap	<p>The project maintains a roadmap that highlights the list of priorities and focus areas. The roadmap is available from https://deeplearning4j.org/roadmap.</p>
License	<p>Apache 2.0</p>
Web Site	<p>http://deeplearning4j.org/</p>

Code Repository	https://GitHub.com/deeplearning4j/deeplearning4j
Twitter	@deeplearning4j
Platforms	Linux, macOS, Windows, Android
APIs	<p>Deeplearning4j can be used via multiple API languages including Java, Scala, Python and Clojure.</p> <ul style="list-style-type: none">• The Scala API is called ScalNet.• Keras serves as its Python API.• The Clojure wrapper is known as DL4CLJ.

Language Breakdown




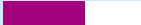





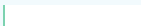





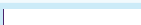

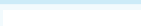
Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Java	190.608	52.492	21.6%	48.115	291.215	 56.6%
 JavaScript	108.988	42.586	28.1%	21.937	173.511	 33.7%
 CSS	32.878	883	2.6%	2.494	36.255	 7.0%
 XML	4.484	944	17.4%	706	6.134	 1.2%
 Scala	2.872	271	8.6%	551	3.694	 0.7%
 HTML	1.764	155	8.1%	218	2.137	 0.4%
 shell script	603	210	25.8%	135	948	 0.2%
 Python	255	18	6.6%	91	364	 0.1%
 Ruby	75	0	0.0%	0	75	 0.0%
Totals	342.527	97.559		74.247	514.333	

Figure 42: DeepLearning4J is written primarily in Java (56.6%) and JavaScript (33.7%)

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

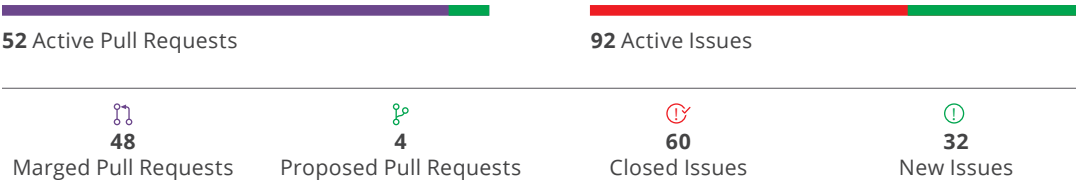


Figure 43: DeepLearning4J - GitHub project pulse (March 15 to April 15)

GitHub Project Insights

Figure 43 illustrates the development activity log using GitHub pulse for 1 month, March 15 to April 15, 2018. Excluding merges, 14 authors have pushed 98 commits to master and 174 commits to all branches. On master, 396 files have changed and there have been 8,124 additions and 5,061 deletions. Finally, 60 issues were closed and 32 new issues emerged.

Observations

The project consists of a very large code base with source code dating as back as 2013. From 2013 to 2016, the project saw significant growth in term of unique contributors, number of patches coming into the projects, and a good mix of corporate, academic and hobbyist. Since 2017, the project is experiencing a decline in terms of contributors and contributors. However, development is still going forward with about 26 unique contributors in 2018.

Figure 44 and Figure 45 from our Facade analysis illustrate respectively the commit history of the project from January 1, 2013 to April 15, 2018 and the number of unique contributors by affiliation.

	2013	2014	2015	2016	2017	2018	Total
Clever Cloud Computing	78	958	0	0	0	0	1.036
(Unknown)	0	50	1.257	2.471	2.316	469	6.563
SkyMind Deep Learning	0	0	639	208	178	12	1.037
BLIX	0	347	371	0	0	0	718
(Hobbyist)	0	46	0	0	53	0	99
Akalea	0	0	8	0	0	0	8
Intel	0	0	31	0	0	0	31
(Academic)	0	0	21	6	10	3	40
Nokia	0	0	0	2	11	0	13
DevFactory	0	0	0	11	0	0	11
Komfo	0	0	0	0	3	8	11
Aurea	0	0	0	8	0	0	8
Ravel Law	0	6	0	0	0	0	6
ORGANIZER	0	0	4	0	0	0	4
Vinted	0	0	1	0	0	0	1
RIPE Network Coordination Center	0	0	1	0	0	0	1
import.io	0	0	4	0	0	0	4
Lightbend Inc.	0	0	0	1	0	0	1

Figure 44: DeepLearning4J - patches by author affiliations (2018 covers January to April 15)

Cloudera	0	0	0	3	0	0	3
Blurb	0	0	2	0	0	0	2
Label Insight	0	0	0	1	0	0	1
NAVER	0	0	0	3	0	0	3
Yandex	0	0	3	1	0	0	4
Seznam	0	0	0	0	5	1	6
SCHUFA	0	0	0	0	0	2	2
e-Trolley	0	0	0	0	2	0	2
Bloomberg	0	0	0	0	0	3	3
Apache Software Foundation	0	0	7	0	0	1	8
Yahoo!	0	0	0	1	0	0	1
Gitter	0	0	1	0	0	0	1
Total from all contributors	78	1.407	2.350	2.716	2.578	499	9.628

Figure 44: DeepLearning4J - patches by author affiliations (2018 covers January to April 15)

	2013	2014	2015	2016	2017	2018	Total
Clever Cloud Computing	1	1	0	0	0	0	1
(Unknown)	0	9	36	52	50	18	135
SkyMind Deep Learning	0	0	3	3	5	2	7
BLIX	0	1	1	0	0	0	1
(Hobbyist)	0	1	0	0	1	0	2
Akalea	0	0	1	0	0	0	1
Intel	0	0	1	0	0	0	1
(Academic)	0	0	6	1	3	2	10

Figure 45: DeepLearning4J - unique contributors by affiliation (2018 covers January to April 15)

Nokia	0	0	0	1	1	0	1
DevFactory	0	0	0	1	0	0	1
Komfo	0	0	0	0	1	1	1
Aurea	0	0	0	1	0	0	1
Ravel Law	0	1	0	0	0	0	1
ORGANIZER	0	0	1	0	0	0	1
Vinted	0	0	1	0	0	0	1
RIPE Network Coordination Center	0	0	1	0	0	0	1
import.io	0	0	1	0	0	0	1
Lightbend Inc.	0	0	0	1	0	0	1
Cloudera	0	0	0	1	0	0	1
Blurb	0	0	1	0	0	0	1
Label Insight	0	0	0	1	0	0	1
NAVER	0	0	0	1	0	0	1
Yandex	0	0	1	1	0	0	1
Seznam	0	0	0	0	1	0	1
SCHUFA	0	0	0	0	0	1	1
e-Trolley	0	0	0	0	1	0	1
Bloomberg	0	0	0	0	0	1	1
Apache Software Foundation	0	0	1	0	0	1	2
Yahoo!	0	0	0	1	0	0	1
Gitter	0	0	1	0	0	0	1
Total from all contributors	1	13	56	65	63	26	181

Figure 45: DeepLearning4J - unique contributors by affiliation (2018 covers January to April 15)

Keras

Project Creator	François Chollet (Now at Google as AI Researcher)
Description	<p>Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.</p> <p>Use of Keras is encouraged if you need a deep learning library that:</p> <ul style="list-style-type: none"> • Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility). • Supports both convolutional networks and recurrent networks, as well as combinations of the two. • Runs seamlessly between CPU and GPU.
Current Status	<p>Active development.</p> <p>Last release was version 2.1.5 on March 6, 2018.</p>
Releases	The project has gone through 40 releases since it was hosted on GitHub. It offers release notes that highlight API changes and offers pointers to areas that require improvements.
Roadmap	The project does not maintain a formal roadmap that captures target features and functionalities for future releases. The project had an effort to maintain a roadmap via “Iteration Plans” published via the GitHub wiki, however, it does not look that it is being put to use. We also noticed that some planning is actually captured in GitHub issues.
License	MIT

Web Site	https://keras.io/
Blog	https://blog.keras.io/
Code Repository	https://GitHub.com/keras-team/keras
Platforms	Linux, macOS, Windows
APIs	Keras supports R and Python interfaces.

Language Breakdown





Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Python	42.483	15.763	27.1%	11.517	69	 100.0%
 Make	22	0	0.0%	7	29	 0.0%
Totals	42.505	15.763		11524	69.792	

Figure 46: Keras is written in Python

March 15, 2018 - April 15, 2018

Period: 1 month

Overview

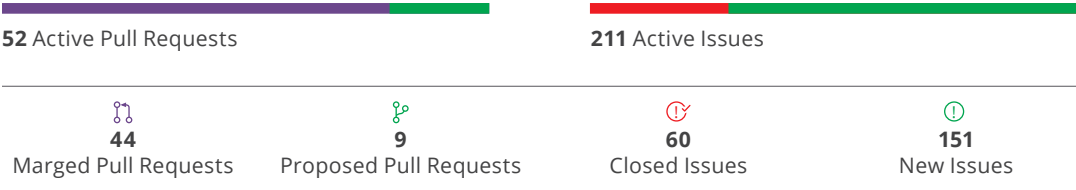


Figure 47: Keras - GitHub project pulse (March 15 to April 15)

GitHub Project Pulse

Figure 47 illustrates the development activity log using GitHub pulse for 1 month, March 15 to April 15, 2018. Excluding merges, 26 authors have pushed 50 commits to master and 50 commits to all branches. On master, 44 files have changed and there have been 1,473 additions and 712 deletions. Finally, 60 issues were closed and 151 new issues emerged.

Observations

Figure 48 displays the commits accepted into the project sorted by author affiliation. The development activities are steady with very comparable number of commits for 2016 and 2017. We also looked at the LoC added and removed during those two years and the numbers are very close. The development activities in 2018, on a month-to-month basis, is comparable to that of previous years, signaling a steady development pace.

	2013	2014	2015	2016	2017	2018	Total
Google	0	0	832	761	486	72	2.151
(Unknown)	0	0	650	465	656	152	1.923
Ola Search	0	0	130	40	24	3	197
(Hobbyist)	0	0	0	1	29	48	78
Microsoft	0	0	0	0	1	0	1
(Academic)	0	0	23	13	39	5	80
IBM	0	0	0	1	6	0	7
Yahoo!	0	0	9	6	3	0	18
Red Hat	0	0	1	0	0	0	1
Yandex	0	0	0	0	1	1	2
Amazon	0	0	2	0	0	0	2
National ICT Australia	0	0	0	1	0	0	1
Intel	0	0	0	0	1	0	1
Senseta	0	0	0	0	0	1	1
Unity	0	0	0	0	1	0	1
CERN	0	0	0	1	0	0	1
Infoblox	0	0	0	0	0	1	1
Total from all contributors	0	0	1.647	1.289	1.247	283	4.466

Figure 48: Keras - patches by author affiliation (2018 covers January to April 15)

Figure 49 provides the number of unique contributors by affiliations. The project has seen a steady increase in number of unique contributors since 2015. For 2018, the number of unique contributors is comparable to the previous year on a month-by-month basis.

	2013	2014	2015	2016	2017	2018	Total
Google	0	0	1	2	2	3	4
(Unknown)	0	0	115	225	296	79	639
Ola Search	0	0	1	1	1	1	1
(Hobbyist)	0	0	0	1	2	1	2
Microsoft	0	0	0	0	1	0	1
(Academic)	0	0	8	11	23	4	43
IBM	0	0	0	1	1	0	2
Yahoo!	0	0	4	4	3	0	10
Red Hat	0	0	1	0	0	0	1
Yandex	0	0	0	0	1	1	1
Amazon	0	0	1	0	0	0	1
National ICT Australia	0	0	0	1	0	0	1
Unity	0	0	0	0	1	0	1
CERN	0	0	0	1	0	0	1
Intel	0	0	0	0	1	0	1
Senseta	0	0	0	0	0	1	1
Infoblox	0	0	0	0	0	1	1
Total from all contributors	0	0	131	247	332	91	711

Figure 49: Keras - unique contributors by affiliation (2018 covers January to April 15)

Apache MXNet

Project Host	Apache Software Foundation
Description	MXNet is an Apache incubator project. It is a deep learning framework used to train, and deploy deep neural networks. It is scalable, allowing for fast model training, and supports a flexible programming model and multiple languages. The MXNet library is portable and can scale to multiple GPUs and multiple machines. Major Public Cloud providers including AWS and Azure support MXNet.
Current Status	Active development. The last release was version 1.1.0 on February 19, 2018.
Releases	The project has gone through 42 releases since it was hosted on GitHub. It offers release notes that highlighting new features, bug fixes, API changes, performance improvements, and any known issues.
Roadmap	For development roadmap, please visit https://GitHub.com/apache/incubator-mxnet/labels/Roadmap .
License	Apache 2.0
Web Site	https://mxnet.apache.org/
Code Repository	https://GitHub.com/apache/incubator-mxnet
Twitter	@ApacheMXNet

Platforms	Windows, macOS, and Linux.
APIs	Python, Scala, R, Julia, C++, and Perl.

Language Breakdown





































Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 C++	127.321	25.205	16.5%	14.032	166.558	 34.0%
 Python	101.430	41.085	28.8%	22.723	165.238	 33.8%
 Scala	27.256	10.164	27.2%	4.560	41.980	 8.6%
 JavaScript	21.754	5.297	19.6%	5.389	32.440	 6.6%
 Perl	15.085	5.321	26.1%	2.805	23.211	 4.7%
 R	10.974	2.713	19.8%	1.662	15.349	 3.1%
 CUDA	10.181	2.962	22.5%	1.632	14.775	 3.0%
 shell script	4.423	2.846	39.2%	1.275	8.544	 1.7%
 XML	3.806	92	2.4%	197	4.095	 0.8%
 CMake	1.960	362	15.6%	344	2.666	 0.5%
 C	1.789	2.549	58.8%	243	4.581	 0.9%
 Make	1.675	631	27.4%	599	2.905	 0.6%
 CSS	1.666	70	4.0%	329	2.065	 0.4%
 Matlab	1.446	462	24.2%	221	2.129	 0.4%
 HTML	663	114	14.7%	74	851	 0.2%
 Java	511	115	18.4%	95	721	 0.1%
 Groovy	420	23	5.2%	16	459	 0.1%
 DOS batch script	375	122	24.5%	119	616	 0.1%
Totals	332.735	100.133		56.315	489.183	

Figure 50: MXNet is written primarily in C++ (34.0%) and Python (33.8%)

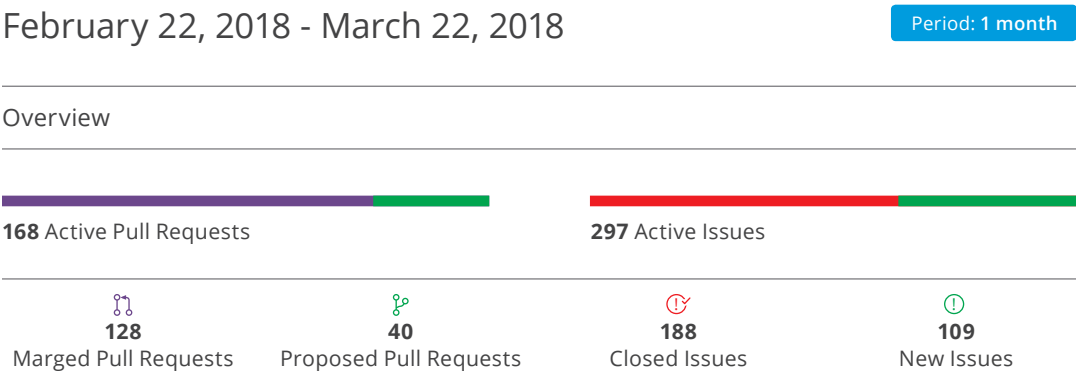


Figure 51: MXNet - GitHub project pulse (February 22 to March 22)

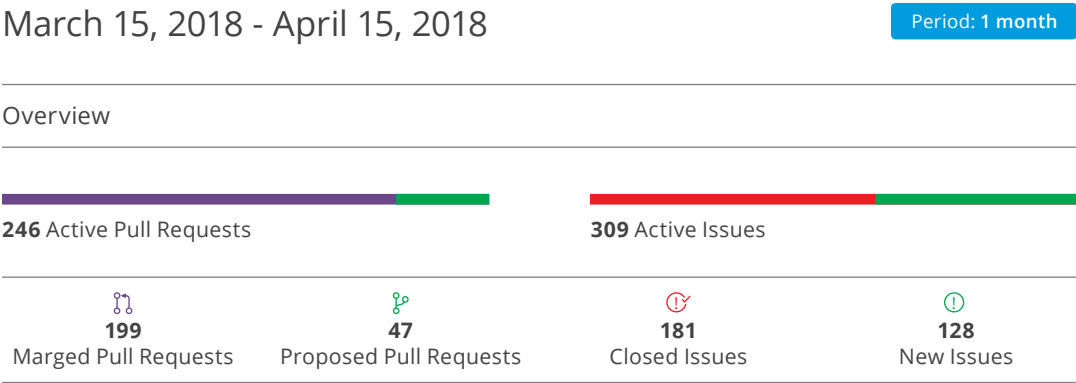


Figure 52: MXNet - GitHub project pulse (March 15 to April 15)

GitHub Project Insights

Figure 51 illustrates a healthy development activity log using GitHub pulse for 1 month, February 22 to March 22, 2018. Excluding merges, 49 authors have pushed 1211 commits to master and 131 commits to all branches. On master, 447 files have changed and there have been 24,330 additions and 5,555 deletions. Finally, 188 issues were closed and 109 new issues emerged.

We came back to the project three weeks after we took that initial GitHub insight screenshot to see if there was progress month over month. Excluding merges, 66 authors have pushed 198 commits to master and 206 commits to all branches. On master, 642 files have changed and there have been 24,644 additions and 6,435 deletions. Finally, 181 issues were closed and 128 new issues emerged.

Observations

The MXNet project is still in the Apache incubator. Despite its relative short history, the project established a solid code base and attracted a large and active development community. Figure 53 shows the number of unique contributors by affiliation.

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	0	0	87	218	268	103	539
(Academic)	0	0	7	19	19	3	36
Intel	0	0	0	2	4	3	6
Amazon	0	0	0	2	6	3	8
(Hobbyist)	0	0	1	0	1	0	2
Comcast	0	0	0	0	1	1	1
Apache Software Foundation	0	0	0	0	2	1	3
Uptake	0	0	1	1	0	0	1
TUPU TECH	0	0	1	1	0	0	1
Tencent Technology	0	0	0	1	0	0	1
SINA	0	0	0	0	1	0	1
Yahoo!	0	0	0	0	2	0	2
Facebook	0	0	0	0	1	1	1
Riverbed Technology	0	0	0	1	0	0	1
Alibaba	0	0	1	0	0	0	1
Microsoft	0	0	1	0	0	0	1
Fujitsu	0	0	0	1	0	0	1
Gitter	0	0	1	0	0	0	1
Total from all contributors	0	0	100	246	305	115	607

Figure 53: MXNet - unique contributors by affiliation (2018 covers January to April 15)

Figure 54 illustrates the number of commits by author affiliation. We can observe a slight decline year-over-year in the number of commits. The reason for this is unclear, and it might signal a warning flag for several possible reasons ranging from a decline in interest in the project to the possibility of having reached a mature state that does not require major developer efforts. The development stats for 2018 are very comparable on a monthly average basis to those of 2017. Therefore, we think the project may be stabilizing and the focus is moving from major development into select new features, code stabilization and bug fixes.

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	0	0	2,008	1,773	1,488	481	5,750
(Academic)	0	0	501	261	109	19	890
Intel	0	0	0	9	16	9	34
Amazon	0	0	0	2	19	20	41
(Hobbyist)	0	0	1	0	46	0	47
Comcast	0	0	0	0	15	5	20
Apache Software Foundation	0	0	0	0	23	5	28
Uptake	0	0	68	8	0	0	76
TUPU TECH	0	0	33	4	0	0	37
Tencent Technology	0	0	0	1	0	0	1
SINA	0	0	0	0	1	0	1
Yahoo!	0	0	0	0	2	0	2
Facebook	0	0	0	0	10	1	11
Riverbed Technology	0	0	0	1	0	0	1
Alibaba	0	0	1	0	0	0	1
Microsoft	0	0	2	0	0	0	2
Fujitsu	0	0	0	1	0	0	1
Gitter	0	0	1	0	0	0	1
Total from all contributors	0	0	2,615	2,060	1,729	540	6,944

Figure 54: MXNet - patches by author affiliation (2018 covers January to April 15)

Apache PredictionIO

Project Host	Apache Software Foundation
Description	An Apache incubator project that offers a machine-learning server for developers and ML engineers. It is built on top of an open source stack (Apache Spark, HBase and Spray) for developers and data scientists to create predictive engines for any machine-learning task.
Current Status	Active development. Last release was version 0.12.0-incubating on September 27, 2017.
Releases	The project has gone through 47 releases since it was hosted on GitHub. The project does not make available detailed release notes. Instead, it provides pointers to the commits that were accepted as part of the releases.
License	Apache 2.0
Web Site	http://predictionio.apache.org/
Code Repository	https://GitHub.com/apache/predictionio
Twitter	@PredictionIO

Language Breakdown













Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Scala	26.813	9.570	26.3%	5.359	41.742	 78.7%
 Python	3.678	1.114	23.2%	1.040	5.832	 11.0%
 shell script	2.437	946	28.0%	565	3.948	 7.4%
 Ruby	814	181	18.2%	197	1.192	 2.2%
 XML	125	31	19.9%	7	163	 0.3%
 Java	109	62	36.3%	22	193	 0.4%
Totals	33.976	11.904		7.190	53.070	

Figure 55: PredictionIO is written primarily in Scala (78.7%) and Python (11%).

Observations

The project is still in incubation at the Apache Software Foundation. The development on the project peaked in 2015 with 76 unique contributor committing 1,294 patches into the code base. Figure 56 presents the number of unique contributors by affiliation. Figure 57 presents the number of commits by contributor affiliation. Since 2015, the activities and the number of unique contributors have been in decline; there are some commercial contributors, however their contributions have dropped significantly since 2015 or completely stopped.

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	4	15	56	23	20	3	109
PredictionIO	2	8	6	3	0	0	8
Salesforce	3	2	0	3	0	0	6
(Hobbyist)	0	4	4	2	0	1	8
Apache Software Foundation	0	0	0	2	6	2	7
OURSKY	0	0	1	0	0	0	1
Heroku (Cloud App Platform)	0	0	0	0	1	0	1
smartBeemo	0	0	1	0	0	0	1
(Academic)	0	3	2	0	1	0	6
Yahoo!	0	0	1	2	1	0	3
BIZREACH	0	0	0	1	0	0	1
EasyCrowd	0	0	0	1	0	0	1
Session Digital	0	0	1	0	0	0	1
CommerceTools	0	0	1	0	0	0	1
InfoQuest Solutions	0	0	0	0	1	0	1
Datameer	0	0	1	0	0	0	1
JCID Web Development	0	0	0	0	1	0	1
TO THE NEW	0	0	1	0	0	0	1
Trove (Market App)	0	0	1	0	0	0	1
Total from all contributors	9	32	76	37	31	6	159

Figure 56: PredictionIO - unique contributors by affiliation (2018 covers January to April 15)

	2013	2014	2015	2016	2017	2018	Total
(Unknown)	22	366	324	110	57	13	892
PredictionIO	90	1.488	784	39	0	0	2.401
Salesforce	540	9	0	12	0	0	567
(Hobbyist)	0	119	150	7	0	1	277
Apache Software Foundation	0	0	0	54	103	5	162
OURSKY	0	0	5	0	0	0	5
Heroku (Cloud App Platform)	0	0	0	0	12	0	12
smartBeemo	0	0	20	0	0	0	20
(Academic)	0	26	2	0	1	0	29
Yahoo!	0	0	3	2	1	0	6
BIZREACH	0	0	0	1	0	0	1
EasyCrowd	0	0	0	1	0	0	1
InfoQuest Solutions	0	0	0	0	1	0	1
Session Digital	0	0	1	0	0	0	1
CommerceTools	0	0	1	0	0	0	1
Datameer	0	0	1	0	0	0	1
JCID Web Development	0	0	0	0	3	0	3
TO THE NEW	0	0	2	0	0	0	2
Trove (Market App)	0	0	1	0	0	0	1
Total from all contributors	658	2.008	1.294	226	178	19	4.383

Figure 57: PredictionIO - patches by author affiliations (2018 covers January to April 15)

Apache SystemML

Project Creator	<p>IBM Research.</p> <p>On June 15, 2015, IBM announced they are open sourcing SystemML as part of IBM's major commitment to Apache Spark and Spark-related projects. Since then, SystemML has been an Apache Incubator project.</p>
Description	<p>IBM developed SystemML to provide the ability to scale data analysis from a small laptop to large clusters without the need to rewrite the entire codebase. Designed to be used with Apache Spark and the machine-learning library MLlib, it makes it possible to write one code base that applies to multiple industries and platforms, allowing developers to customize applications and integrate deep intelligence into their specialized processes.</p> <p>Future developments include additional deep learning with GPU capabilities such as importing and running neural network architectures and pre-trained models for training.</p>
Project History	<p>SystemML originated at IBM Research in 2010 and was submitted to the Apache incubator in November 2015. On May 31, 2017, Apache SystemML has graduated from the Apache Incubator to become a Top-Level Project, signifying that the project's community and products have been well-governed under the ASF's meritocratic process and principles.</p>
Current Status	<p>Active development.</p> <p>Last release was version 1.0.0 on December 12, 2017. Current focus is on version 1.1.0 with release candidate 1.1.0-rc2 made available on March 23, 2018.</p>
Releases	<p>The project has gone through 47 releases since it was hosted on GitHub. There are no detailed release notes. Instead, it provides pointers to the commits that were accepted as part of the releases.</p>

Roadmap	The project maintains a detailed development roadmap available from https://systemml.apache.org/roadmap .
License	Apache 2.0
Web Site	http://systemml.apache.org/
Code Repository	https://GitHub.com/apache/systemml
Twitter	@ApacheSystemML
APIs	Python, Java

Language Breakdown
























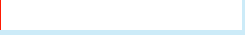



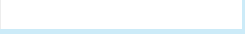



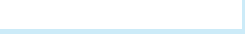


Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
 Java	270.432	77.913	22.4%	55.981	404.326	 85.8%
 Python	11.453	3.963	25.7%	3.013	18.429	 3.9%
 R	8.213	10.773	56.7%	3.699	22.685	 4.8%
 TeX/LaTeX	4.881	252	4.7%	747	5.880	 1.2%
 XML	3.551	467	11.6%	166	4.184	 0.9%
 Scala	3.535	1.432	28.8%	366	5.333	 1.1%
 shell script	1.835	1.560	45.9%	535	3.930	 0.8%
 HTML	1.784	31	1.7%	171	1.986	 0.4%
 CUDA	1.191	584	32.9%	188	1.963	 0.4%
 C++	635	206	24.5%	127	968	 0.2%
 DOS batch script	339	123	26.6%	120	582	 0.1%
 CSS	294	53	15.3%	62	409	 0.1%
 JavaScript	188	53	22.0%	24	265	 0.1%
 CMake	181	84	31.7%	54	319	 0.1%
 C	40	54	57.4%	18	112	 0.0%
 Make	18	40	69.0%	11	69	 0.0%
 Ruby	3	0	0.0%	0	3	 0.0%
Totals	308.573	97.588		65.282	471.443	

Figure 58: SystemML is written primarily in Java (85.8%)

Observations

The project has an established code base thanks to the initial drop from IBM and its continued development effort on the project. However, there are multiple signs of decreasing activities in 2017 and 2018 in comparison with the development activities in 2015 (when the project started as an Apache incubator project) and 2016. Figure 59 shows the number of unique contributors by affiliation and Figure 60 presents the number of commits from authors by affiliation.

	2013	2014	2015	2016	2017	2018	Total
IBM	13	18	23	12	9	3	41
(Unknown)	0	0	4	9	12	4	23
Apache Software Foundation	0	0	2	3	2	2	4
Yahoo!	0	0	1	1	1	0	1
(Academic)	0	0	0	2	2	1	4
Atos	0	0	0	0	0	1	1
(Hobbyist)	0	0	0	2	0	0	2
Capital One	0	0	0	1	1	0	1
Total from all contributors	13	18	30	30	27	11	77

Figure 59: SystemML - unique contributors by affiliation (2018 covers January to April 15)

	2013	2014	2015	2016	2017	2018	Total
IBM	407	707	1,344	667	378	29	3,532
(Unknown)	0	0	9	47	609	234	899
Apache Software Foundation	0	0	37	44	36	4	121
Yahoo!	0	0	12	33	62	0	107
(Academic)	0	0	0	4	14	1	19
Atos	0	0	0	0	0	3	3
(Hobbyist)	0	0	0	2	0	0	2
Capital One	0	0	0	4	1	0	5
Total from all contributors	407	707	1,402	801	1,100	271	4,688

Figure 60: SystemML - patches by author affiliation (2018 covers January to April 15)

The project was able to attract some contributions from corporate developers but these contributions declined over the years and in some cases completely stopped. IBM remains the most active corporate contributor.

The Linux Foundation AI Efforts

On March 26, 2018, The Linux Foundation launched the [LF Deep Learning Foundation](#), an umbrella organization that will support and sustain open source innovation in artificial intelligence, machine learning, and deep learning while striving to make these critical new technologies available to developers and data scientists everywhere. LF Deep Learning was created to support numerous technical projects within this important space. With LF Deep Learning, members of the projects are working to create a neutral space for harmonization and acceleration of separate technical projects focused on AI, ML, and DL technologies.

The goal of the Linux Foundation from hosting the Acumos AI platform and the Acumos Marketplace is to nurture an active, large ecosystem around the project to sustain it over the long term.

Project Name	Acumos AI
Project Host	LF Deep Learning Foundation
Description	<p>Acumos AI is a platform and open source framework that makes it easy to build, share, and deploy AI apps. Acumos standardizes the infrastructure stack and components required to run an out-of-the-box general AI environment. This frees data scientists and model trainers to focus on their core competencies and accelerates innovation.</p> <p>As such, the Acumos AI Platform is a complete environment for the full lifecycle of AI and ML application development.</p>
Acumos AI Marketplace	The free Acumos Marketplace packages various components as microservices and allows users to export ready-to-launch AI applications as containers to run in public clouds or private environments. https://marketplace.acumos.org
Current Status	Project launched on March 26, 2018.

License	Apache 2.0
Web Site	https://www.acumos.org/
Blog	https://www.acumos.org/blog/
Code Repository	https://gerrit.acumos.org/r/#/admin/projects/
Twitter	@AcumosAI
APIs	The Acumos AI Platform offers APIs to connect and chain models and toolkits as microservices.

In addition to the Acumos AI Project, LF Deep Learning anticipates future project contributions from Baidu and Tencent, among others:

- Baidu's EDL project enhances Kubernetes with the feature of elastic scheduling and uses PaddlePaddle's fault-tolerable feature to significantly improve the overall utilization of Kubernetes clusters.
- Tencent's Angel project, a high-performance distributed machine-learning platform jointly developed by Tencent and Peking University, is tuned for big data/models

We will feature the efforts of the LF Deep Learning Foundation in a separate publication in the near future.

SECTION III

Observations

Common Characteristics

Development and governance is dominated by a single large entity

In almost all projects (hosted outside an open source foundation), the most popular projects are heavily influenced by a single large entity such as Google, Facebook, IBM, etc. We believe that one of the contributing factors to this phenomenon is that AI platform development requires a very narrow band of specialized knowledge. Most AI platforms are the results of years of investment and talent acquisition, and the open source spinoff is a consequence of wanting to build an ecosystem versus a desire to collaborate with others on constructing a platform.

Developed internally to address specific product requirements

Due to the cost of human capital and the time required to develop such a complex technology, AI systems are usually created with a specific product or service in mind. As such, platform development is more likely to follow the *source-available* pattern of open source contribution, as compared to open platform development in Linux. In this model, plans and code are developed internally and periodically published for the ecosystem for review and consumption. External contributors may propose code, but generally, the project host (who employs the maintainers of the project) is under no obligation to treat external contributors as equal to internal contributors. This is not specific to open source AI projects as we see it in many projects across domains.

Highly specialized to perform certain types of tasks

One significant consequence of this type of development is that users of the code benefit greatly when their use cases overlap with the host's (for example, a company using TensorFlow for image comprehension) because it provides them with a consistent stream of product-ready code given Google's heavy involvement in the project. However, it becomes more challenging when a user plans to adapt the host's code. At this point, the typical methods used to influence upstream open source projects become less effective. In conclusion, there is a necessary tradeoff between the completeness of the platform and its adaptability. This is generally mitigated by and open governance that allows contributors to broaden the project's scope, but few AI projects have achieved this level of open governance with a defined project structure, processes, and clear ways to promote developers to committers or maintainers.

Little contributor cross-pollination

Since most large companies (Microsoft, Facebook, Apple, IBM, etc.) have open sourced their own AI related projects (platform code, frameworks, libraries), they are all focusing on these projects with little to no contributions to other projects. This situation has created a very limited cross-pollination across projects and has led most projects to have one or two major players at best.

Most projects are tightly coupled with their original authors

One of the striking observations was that most projects are tightly coupled with their original authors. When the original authors move on to other projects because of changing affiliations (i.e. having a new employer, or leaving academia to industry), the projects slowdown drastically, and in many cases, development has halted in favor of other AI projects supported by the new employer. This situation would have been preventable with strong contributor cross-pollination across projects; however, due to the nature of the AI open source projects, this is proving to be hard to exercise.

Fast Release Cycles Dominate Development

One noticeable feature among the surveyed projects was the frequency of their releases and the speed of development. These projects took the “release early, release often” open source development philosophy to heart. They emphasized the importance of early and frequent releases in creating a tight feedback loop between developers, testers and users.

Major Improvements in Documentation

One very positive surprise was the abundance of documentation across all these projects. Lack of documentation has always been an issue with open source projects and the situation is improving gradually. However, most likely, due to the specialization requirement in the AI domain, these open source projects come with a great amount of user and development documentation, tutorials, detailed examples, test cases, sample data sets, API documentation, with some projects even providing video tutorials.

Academia and AI

The overnight academic success with AI research is really the result of decades of hard work, government funding, and tireless professors and students who kept on going despite the various challenges. Open source creates a great environment for the academic world to collaborate with the industry without imposing too many restrictions on that type of relationship.

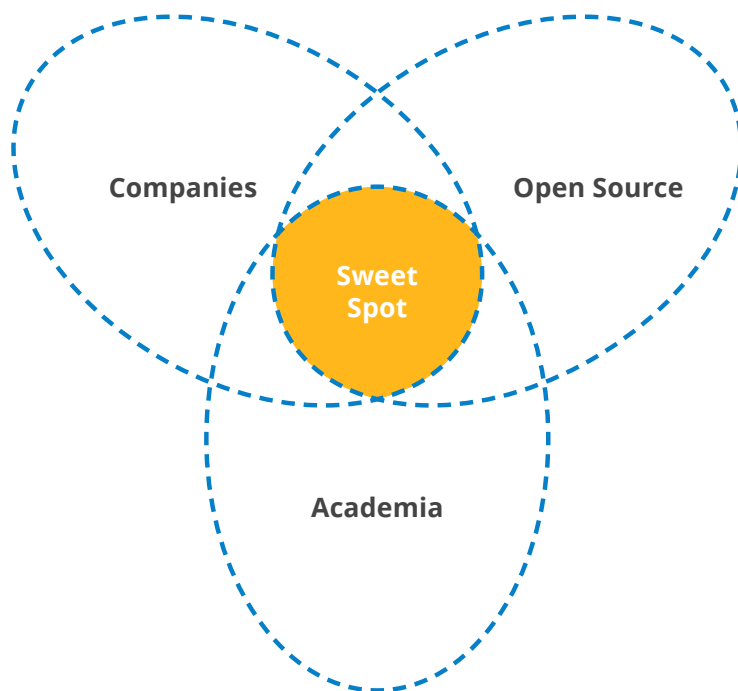


Figure 61: Finding the sweet spot – $\text{Companies} \cap \text{Open Source} \cap \text{Academia}$

The sweet spot is the intersection in interest from the academia, the open source community and enterprises. In recent years, interest in AI started to grow incredibly fast in enterprises at a time when AI has been an established R&D domain in Academia with reference implementations to many of the ideas already existing as open source projects. The interests aligned and today the rest is history.

For instance, the appetite from companies to open up AI R&D labs in Canada or capture existing Academic labs under their sponsorship in cities like Montreal, Toronto, and Vancouver, is driven by the eagerness to capture the abundance of Canadian AI talent. This talent is the result of decades of government and tax payers support in that domain combined with the use of open source as a tool to collaborate with other academics on the implementation side.

Many similarities are worth noting between Academia and open source:

- Their collaborative approach to innovation.
- The process of creating and validating new ideas.
- The licenses applied to the source code resulting from the R&D.
- Community building – although they build communities in different circles, they do share that passion and practice.
- The importance of sharing results at conferences and getting external reviews and feedback.

In the AI field, companies joined the party and they were able to put in an incredible amount of resources into the space and improve on the general Academic approach and practices in areas such as:

- The finality of produced software,
- The approach to architecture (plugins, APIs),
- Accelerating the inception of initiatives,
- Building a developer ecosystem around these projects,
- Gathering contributions for the projects from other companies (typically business partners),
- Providing access to large data sets that can be used for training specific models,
- Improving documentation, developer tools, and developer support in general,
- Increasing API coverage and support, and
- Applying these technologies to products and services.

The result was the culmination of efforts between Academia and enterprises under an open source umbrella that created the right conditions for such a collaboration to happen.

Many of the AI related open source project, platform, framework and libraries started as academic R&D initiated at different universities. For this study, we collaborate with Teqmine to mine over 12.5 million issued patents in the US and the EU. We found that academic institutions were granted 8,597 AI patents for the period extending from January 1, 2006 to December 31, 2017.

The top 10 academic patentees sorted by highest number of issued patents are:

- California University System
- Massachusetts Institute of Technology
- Columbia University
- California Institute of Technology
- Johns Hopkins University
- Stanford University
- University of Michigan
- University of Texas
- Hong Kong University of Science and Technology
- New York University

The very interesting part is the realization that some of these academic institutions have generated more AI patents than many enterprises that fall under the Fortune 100 Technology Companies.

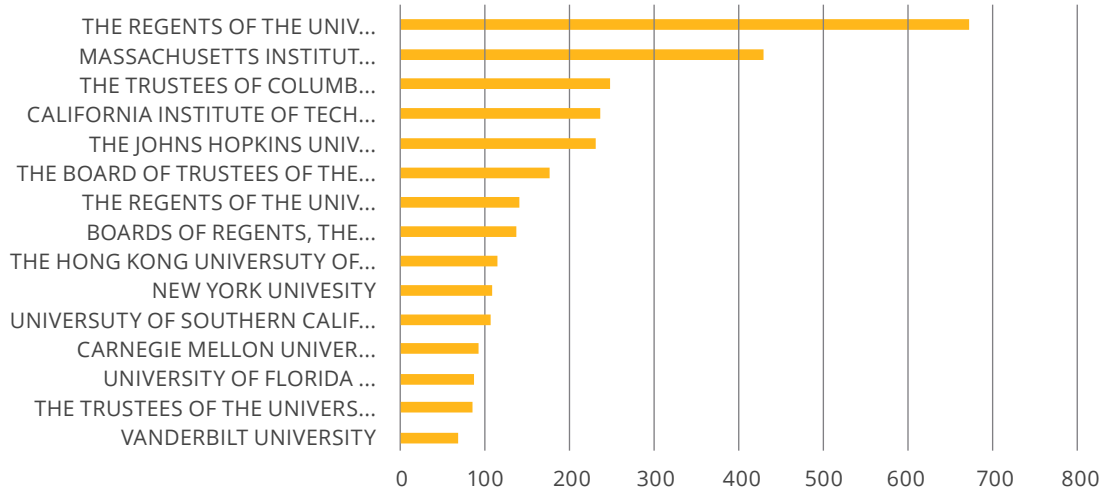


Figure 62: Top 15 Academic AI patents Assignees (2006-2017) - Source: Teqmine.com

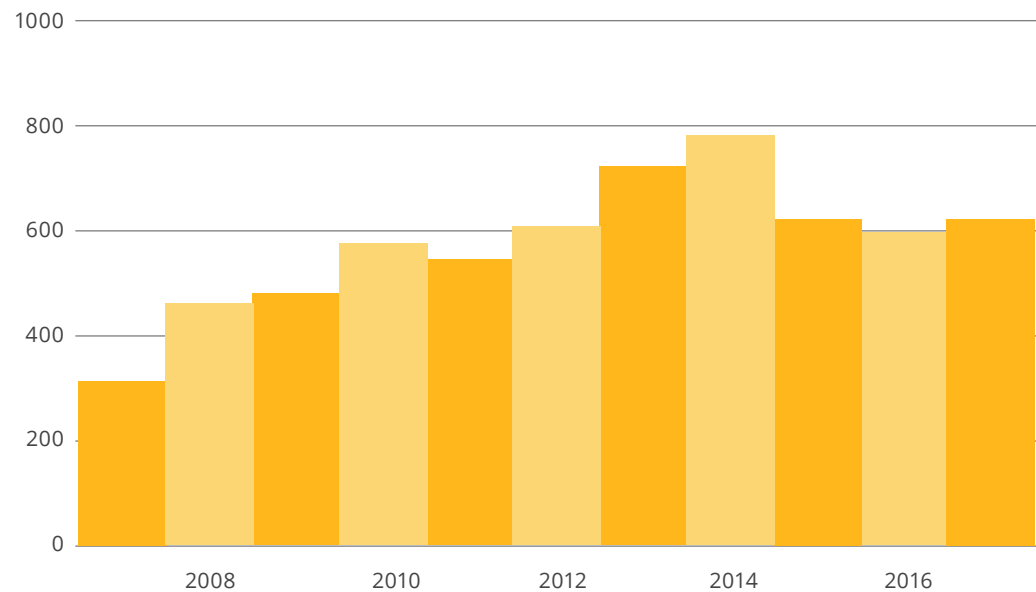


Figure 63 provides the illustration showing the yearly AI patenting activity from academic institutions covering the period from January 2006 and ending in December 2017.

Figure 63: Number of issued AI patents to Academic institutions (2006-2017) - Source: Teqmine.com

Open Source and AI

While most AI productization is still done internally and is a source of proprietary advantage, it is increasingly common to see AI as open source projects. This is in part due to its roots in academia, which has historically been a steady source of open source proof-of-concept projects. However, it is also rooted in the cost to build a platform and the realization that the true value is in the models, training data, and the apps. By open sourcing a platform, the host has an opportunity to recruit others (possibly unpaid) contributors, particularly if other organizations can be incentivized to integrate the platform into their own products. As with any technology where talent premiums are high, the network effects of open source are very strong.

Exodus from Academia

Given the strong coupling of AI research and academic R&D labs, the current wave of AI talent hiring has pushed many AI researchers out from academic research into commercial and product R&D. There have been even cases of companies sponsoring and taking complete AI labs from various academic institutions under their employment, bringing more money in R&D sponsorships to those academic institutions, and nearly monopolizing these labs for the benefit of their own AI efforts.

Incubators Work

Many of the projects we have looked at started or are currently incubated at the Apache Software Foundation, which has a varied focus that includes platforms and algorithms. This has proven how valuable incubators can be as testing grounds for ideas. Some projects pass the incubator [graduation requirements](#), succeed, and graduate. Projects that do not meet these requirements remain in incubation. One great example is Apache Spark, which, over the past twelve months, received contributions from over 400 developers affiliated with more than 80 companies and several academic institutions.

The projects that do not graduate are not really a failure in any measure. We believe that these projects offer a great learning experience, increase knowledge, and offer a pivoting opportunity for the project participants into the new ideas that learn from previous experience to create a better approach.

GitHub is Dominant

All projects we surveyed had a presence on GitHub, a strong confirmation of the dominant position GitHub maintains in the space of enabling collaborative development.

Open Source Licenses and Open Training Data

Most of the time, open source licensing discussions are about source code. In the case of the projects surveyed, the licenses all of the projects adopted are OSI-approved open source licenses. This means developers are familiar with these licenses and in general, the end users and adopters of the software are familiar with the license and their obligations. Put differently, no successful projects use custom open source licenses, and the adoption of OSI-approved open source licenses keeps with industry best practices that strongly discourage customization, particularly related to patent grants.

However, building a meaningful AI application requires both a model and data with which to train it. These datasets are generally large and specially designed to effectively train models to draw certain types of conclusions.

As much as open source licenses helped commoditize certain types of source code, open data licenses are beginning to commoditize training data. Licenses such as [Community Data License Agreement \(CDLA\)](#) provide a structured set of guidelines to enable open sharing of datasets under defined terms. The availability of training data under these terms will help democratize the overall AI marketplace by lowering the barriers to entry when offering an AI-backed service. Proprietary datasets will continue to exist of course, but the availability of data under these new licenses should make it possible for everyone to build credible products, including the smaller players.

Consolidation, Winners and Winners

We expect consolidation around multiple platforms and libraries that address the same problem. This is happening now, in fact, with some projects halting development as contributors flock to other competing projects. However, unlike other consolidation scenarios, we do not believe the result is a contest of winners vs. losers. The outcome is more likely to be win-win as successful projects win by grabbing their share of contributors, and contributors (including those from Academia) gain knowledge, experience and skills that make them a highly desired workforce. Projects must strive to experiment in the ongoing stream of new ideas and be a playground for academics that teach the open source model and the specific domain expertise to new participants.

Governance Matters

Open source AI projects have a wide variety of governance models. Some projects are tightly coupled to a single host; others such as the Apache-hosted projects have a more diverse contributor base with a broader focus. Based upon historical patterns, open source projects with more diverse contributor patterns last longer and are more likely to survive when a major contributor changes strategy; this is logical because other contributors can fill in vacuums in leadership before the project collapses. It is reasonable to conclude that projects like TensorFlow and Caffe2 benefit greatly from the focused momentum of a single large host, but that the lack of contributor diversity is a continuous existential risk for the projects.

Open Source Development and Collaboration Model

Open source has been eating the software world across multiple industries. Today, more leading-edge software development occurs inside open source communities than ever before, and it is becoming increasingly difficult for proprietary projects to keep up with the rapid pace of development that open source offers. AI is no different and is becoming dominated by open source software that brings together multiple collaborators and organizations. One key question in that regard is what makes open source an attractive development and collaboration model for AI. The characteristics of the model of open source make it an ideal environment to collaborate on enabling technologies regardless of domain or industry. In this section, we summarize what makes open source a special environment that fosters collaboration and innovation.

Access to a Larger User and Developer Ecosystem

Larger open source communities consist of a wide range of developers and users across multiple companies, industries, and goals; they can provide value to your organization that you wouldn't be able to support otherwise through internal efforts alone. This includes adaptations to new and fringe use cases, translations to other languages, developer and user support, and more. More importantly, when other external participants and organizations contribute engineering effort to an open source project, all organizations that participate get to leverage these benefits without investing the time to create them.

Improved Code Quality and Stability

In an open source community, anyone is free to report, test, and fix bugs, regardless of their association; this often results in a faster identification and resolution of problems like bugs and security issues. Additionally, the processes a community uses for this identification and resolution can be evaluated to determine if they meet the risk profile of your specific company.

Create Technical and Political Leadership

Leadership in an open source community allows a company to ensure important software remains viable for the company while ensuring the company has a say in community decisions. Open source leadership must be earned through ongoing participation and can be lost due to a lack of participation so it is vital to support ongoing efforts. Participating companies benefit from this by growing internal expertise that can better leverage innovative open source AI technologies in the company's products and services.

Reduces Licensing Costs

Open source AI provides simplified licensing that eliminates initial licensing costs, producing savings in the software procurement process since open source software can simply be downloaded for experimentation, research, and use in products and services. Open source licenses also allow a much broader range of customization without the need to negotiate contracts with third parties.

Collaboration

Successful open source communities bring together participants across multiple organizations and industries to collaborate on shared technology. This produces software that is adapted to a wide range of use cases and results in software that is modular to facilitate successful collaboration.

Faster Innovation

The biggest benefit open source provides is the potential to be modified whenever necessary without the need to negotiate custom contracts with third parties; this alone can result in significant improvements to development speed. Open source communities also produce a faster evolution of upstream software that is based on the needs of the community rather than the financial motivations of a particular company.

Faster Time to Market

Open source AI platforms provide better control of the software stack by means of a relatively unlimited potential for customization and better access to the software knowledge base. This allows companies to incorporate it more quickly into their products and services and bring them to market much faster.

Appendix A – Methodology of Contribution Analysis

Tool

We used **Facade** to compile all of the development statistics presented in this paper. Facade is a program that analyzes the contributors to git repos, organized into groups called projects, on a commit-by-commit basis. For more advanced analysis, the tool offers the capability to export the contributor data as a CVS file, giving its users additional freedom in using the data.

Characterization of the Results

- “\$CompanyName” – Contributions coming to projects under a company email address (for instance contributions from IbrahimHaddad@samsung.com) are categorized and counted as Samsung’s contributions.
- “Unknown” – This category corresponds to contributors using private email addresses that are not trackable to their real identify. An example would be ai.superstar@gmail.com. We encountered thousands of such contributors using Gmail, Hotmail, Outlook, and many other localized email service providers. Since we lack the ability to pinpoint the contributor or their affiliation, we group them as Unknown.
- “Academic” – This category includes all contributions coming from academic domains such as *.edu or *.ac.uk.
- “Hobbyist” – This category includes contributions from individuals with a known email identity and the individual is contributing outside the scope of their work responsibilities. In those cases, we list these individuals as “Hobbyist”. If their email identity were unknown, they would fall under the “Unknown” category.

Development Statistics

As with open source projects and their fast development and release cycles, the statistics provided are limited to the specific said period. We highly recommend that readers download and use Facade to track the development of their favorite open source projects.

Acknowledgments

OpenHub – The language breakdown sections for all projects are sourced from openhub.net.

GitHub – The project insight sections are sourced from [GitHub.com](https://github.com).

Teqmine – The AI patent data covering Academia (2006-2017) is sourced in collaboration with Teqmine Analytics Oy ([Teqmine.com](https://teqmine.com)) based in Helsinki, Finland. Teqmine offers automated and AI assisted SaaS solutions for global patent monitoring and patent similarities, in addition to a number of specialized research services.

Feedback

The author apologizes in advance for any misrepresentation of any of the projects and is grateful to receive corrections and suggestions for improvements via ibrahimatlinux.com/contact.html. We will ensure that future revisions of this paper will include your feedback with proper attributions.

About the Author



Ibrahim Haddad is VP of R&D and the Head of the Open Source Group at Samsung Research America. He is responsible for overseeing Samsung's open source strategy and execution, R&D collaborations, supporting M&A activities, and representing Samsung in open source foundations.

Haddad received his with a Ph.D. in Computer Science from Concordia University (Montreal, Canada).

He is the author of **"Open Source Compliance in the Enterprise"** which provides a practical guide on how to best use open source code in products in a legal and responsible way. His latest publication, **"Open Source Audits in Merger and Acquisition Transactions"**, provides a guide to open source audits in M&A transactions, and offers guidelines to improve open source compliance preparedness.

Twitter: [@IbrahimAtLinux](https://twitter.com/IbrahimAtLinux)

Web: IbrahimAtLinux.com



The Linux Foundation promotes, protects and standardizes Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation or our other initiatives please visit us at www.linuxfoundation.org